LIVEASSISTANT: LEARNING WHETHER, WHEN, AND WHOM TO ASSIST IN REAL-WORLD LIVE SOCIAL STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Livestreams are a primary form of online video: long-lasting, interactive environments in which omni-modal content, viewers, hosts, and platform signals unfold together. Yet how AI systems should *act* within them remains largely unexplored. **The central question is how a model can move beyond perceiving what is happening and deciding when to respond proactively, toward selective participation in a shared social environment.** We introduce LIVEASSISTANT, a framework for mixed-initiative, role-conditioned assistance that formulates livestream assistance as four coupled decisions over a multi-party stream: **whether to act, when to act, whom to address, and what to communicate.** At each incoming chunk, LIVEASSISTANT integrates native audio and video with synchronized comments, gifts, viewer dynamics, and room metadata, and chooses to continue observing, update private memory, or produce a grounded response for viewers or hosts. To learn this policy, we build a streaming trajectory engine that reconstructs replay timelines into time-aligned decision trajectories, yielding over 320 hours of training data and a 39-hour human-verified benchmark spanning 8 regions and 7 languages. We further introduce Marker-Aware Multiturn Supervised Fine-Tuning (MA-MSFT) and Streaming Multiturn GSPO (SM-GSPO) to strengthen structured decisions and improve behavior over evolving streams. LIVEASSISTANT attains the strongest joint performance in timing, routing, and content quality among evaluated models, moving streaming assistance from reactive responses toward selective participation in live social environments.

1 INTRODUCTION

Streaming video understanding requires a model to perceive and reason over content as it unfolds. Recent systems have progressed from describing what happens in a continuous input (Chen et al., 2024; Huang et al., 2025; Niu et al., 2025) to deciding when a response should be produced (Qian et al., 2025; Wang et al., 2025a; Yang et al., 2025b). In both paradigms, the interaction is passive or bound by predefined triggers and event rules, lacking the initiative to interpret environmental signals and autonomously decide on interaction.

Livestreaming exposes a more complex interaction problem. Liveroom is a shared social environment in which hosts and viewers participate in the same evolving stream, yet their assistance needs differ. For viewers, the model may answer questions that the host misses; for hosts, it may surface audience confusion or shifts in attention. Deciding whether and when a system should take the initiative is the classical problem of mixed-initiative interaction (Horvitz, 1999; Allen et al., 1999); livestreaming further requires that this initiative be routed across participants with different needs. The model must therefore infer emerging needs from audiovisual content, comments, gifts, and room dynamics (Qu et al., 2026; Guo et al., 2026; Wang et al., 2026), and jointly decide **whether** to act, **when** to act, **whom** to assist, and **what** to communicate.

Existing work covers complementary parts of this problem. **Streaming video models** support online reasoning and response timing, but still assume an externally specified query or trigger (Chen et al., 2024; Wang et al., 2025a; Yang et al., 2025b; Wang et al., 2025c; Tian et al., 2026; Xie et al., 2026; Xu et al., 2026). **Livestream systems** exploit social signals but are specialized for predefined

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.

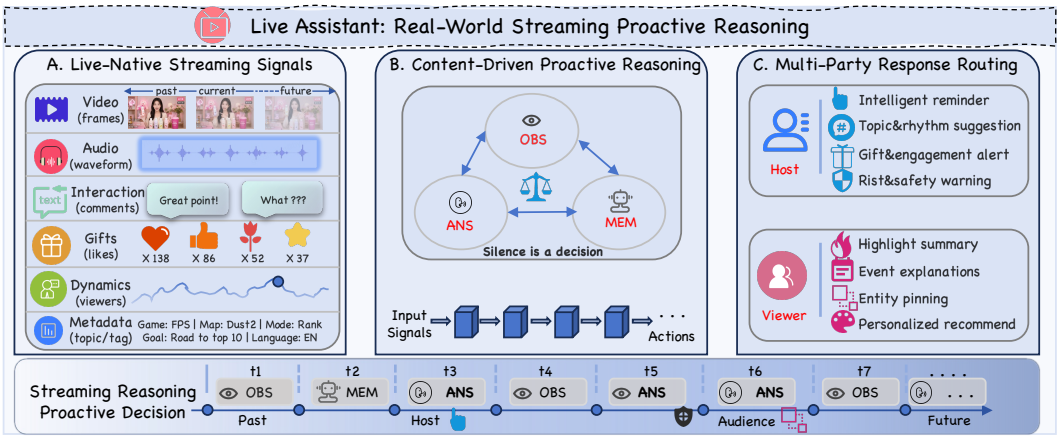


Figure 1: LIVEASSISTANT runs directly on a live social stream. It jointly decides whether and when to act, which participant should receive the action, and what information should be communicated.

functions (Ma et al., 2019; Lalanne et al., 2023; Wang et al., 2026; Qu et al., 2026; Tang et al., 2022; Yew et al., 2026), while **multi-party human-AI interaction** studies recipient selection mainly in turn-based dialogue (Houde et al., 2025; Inoue et al., 2025). The streaming policy that discovers when assistance is needed and routes it to the appropriate participant within the context evolves is still under-explored.

We introduce LIVEASSISTANT, a unified policy for mixed-initiative, role-conditioned assistance in livestreams. At each step, it processes native audio and video together with synchronized comments, gifts, viewer dynamics, and room metadata, and chooses to continue observing (OBS), update private memory (MEM), or emit an external response (ANS). For each response, the same policy determines the recipient, task, and grounded message content. Through this shared action space, LIVEASSISTANT jointly resolves **whether**, **when**, **whom**, and **what** as the stream evolves.

Learning this policy raises three challenges. First, no native supervision specifies per-step state, recipient, task, and grounded response from the evolving multimodal and social context. We therefore build a livestream trajectory engine that reconstructs replay timelines, aligns audiovisual content with platform events, and converts continuous sessions into structured decision trajectories, yielding over 320 hours of optimization data and a human-reviewed 39-hour benchmark. Second, the key control decisions occupy only a few state, recipient, and task markers, while response content spans many more tokens, so standard token-level loss can underweight the decisions that govern behavior. We introduce **Marker-Aware Multiturn Supervised Fine-Tuning** (**MA-MSFT**), which strengthens structured decision supervision while preserving open-vocabulary generation. Third, each action becomes part of the context for subsequent decisions, allowing early errors to propagate through the stream. We therefore introduce **Streaming Multiturn Group Sequence Policy Optimization** (**SM-GSPO**), which trains on on-policy multiturn rollouts with turn- and trajectory-level credit assignment to improve timing, routing, restraint, and non-repetition (Zheng et al., 2025).

Together, these components move streaming multimodal assistance from responding to explicit requests toward selective participation in a shared social environment.

Our contributions are threefold.

- We formulate livestream assistance as a mixed-initiative decision problem, in which a single policy discovers the need for assistance and jointly resolves the four coupled decisions.
- We construct structured livestream trajectories from native audio, video, comments, and platform signals, together with a human-reviewed benchmark for evaluating action timing, recipient routing, task selection, grounding, and response quality under streaming inputs.
- We develop MA-MSFT and SM-GSPO for learning structured decisions in continuous streams. MA-MSFT strengthens sparse decision supervision, while SM-GSPO trains on self-generated trajectories to improve both turn-level decisions and long-horizon behavior.

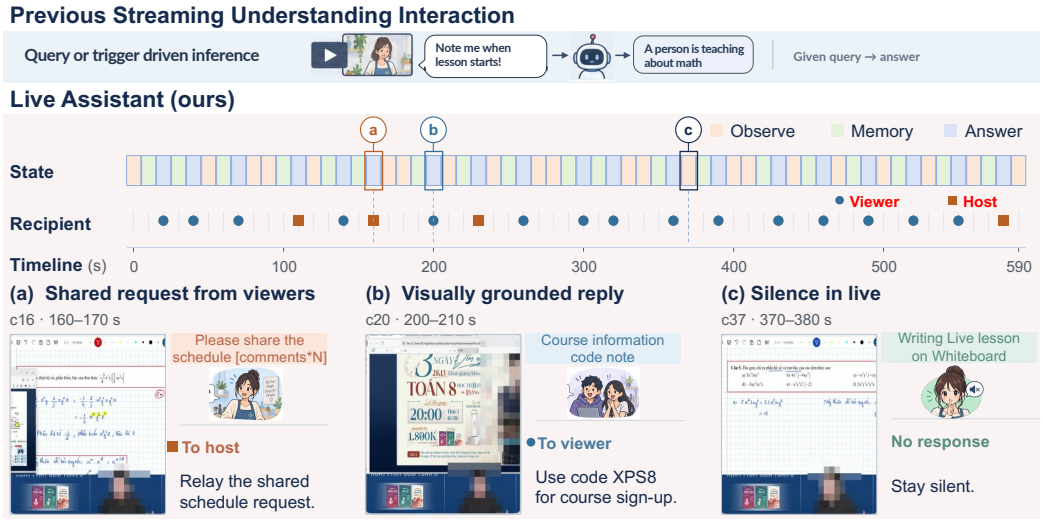


Figure 2: **Selective assistance over a continuous livestream.** In this 10-minute case, an online mathematics instructor writes solutions on a digital whiteboard and responds to viewer questions.

2 Problem Formulation

2.1 Formalization

Context over control. Existing streaming video models typically rely on an explicit query or predefined trigger that specifies what to answer or monitor (Chen et al., 2025; Ding et al., 2025). Our formulation instead uses a persistent generic instruction that defines the assistant's role without specifying a concrete request (given in Section B.4.2). At each step, the model must infer whether an unmet need has emerged, whom to assist, and how to respond. We call this formulation *context over control*: assistance is initiated by needs inferred from the evolving social context rather than by an explicit command. Figure 2 illustrates this distinction through a continuous mathematics livestream in which assistance emerges selectively from the stream context.

We represent a livestream as

$$\mathcal{X} = \{x_t\}_{t=1}^{T}, \qquad x_t = (v_t, a_t, e_t), \tag{1}$$

where v_t and a_t denote native video and audio, and e_t contains synchronized comments, gifts, viewer dynamics, and room metadata. At step t, the model conditions on

$$h_t = (p, x_{\le t}, y_{<t}), \tag{2}$$

where p is the persistent instruction and y_i is the action at step i.

Given h_t, the model predicts a state

$$z_t \in \mathcal{S} = \{\mathrm{OBS}, \mathrm{MEM}, \mathrm{ANS}\}, \tag{3}$$

where OBS produces no message, MEM writes a private contextual update u_t, and ANS emits a set of external messages $\mathcal{M}_t$. Each message is represented as

$$m_{t,k} = (r_{t,k}, q_{t,k}, w_{t,k}), \tag{4}$$

where $r_{t,k} \in \{\mathrm{viewer}, \mathrm{host}\}$ is the recipient, $q_{t,k}$ is the task, and $w_{t,k}$ is the grounded response. The action is serialized as

$$y_t = \begin{cases} \langle \mathrm{obs} \rangle, & z_t = \mathrm{OBS}, \\ \langle \mathrm{mem} \rangle u_t \langle / \mathrm{mem} \rangle, & z_t = \mathrm{MEM}, \\ \langle \mathrm{ans} \rangle \{\langle r_{t,k} \rangle [q_{t,k}]\, w_{t,k}\}_{k=1}^{|\mathcal{M}_t|} \langle / \mathrm{ans} \rangle, & z_t = \mathrm{ANS}. \end{cases} \tag{5}$$

In Figure 2, LIVEASSISTANT routes a shared schedule request to the host, answers viewers when the course code becomes visually available, and remains silent during routine whiteboard instruction. **A single policy jointly determines whether and when to act, whom to assist, and what to communicate.**

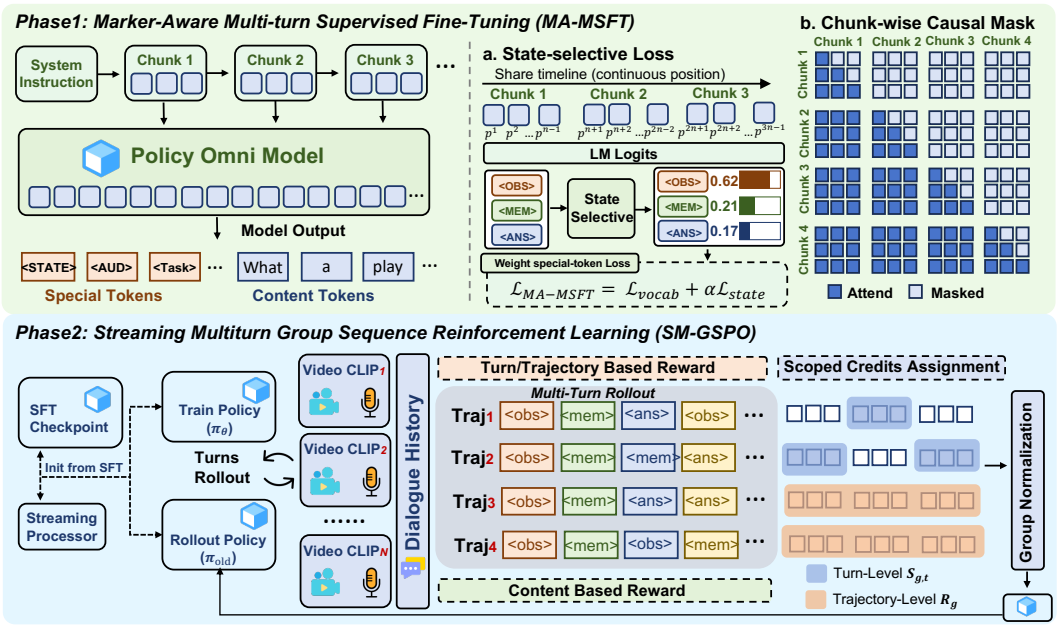


Figure 3: Overview of the LIVEASSISTANT training pipeline.

2.2 CHALLENGES FROM THE FORMULATION

Three properties of Equation 5 are absent from ordinary streaming and question-answering setups, and each one dictates a component of our approach. **(i) Supervision does not exist.** No corpus pairs native livestream media with per-chunk state, recipient, task, and grounding, so the trajectories have to be constructed (Section 3.1). **(ii) The decision is rare structure inside long generation.** Most chunks call for silence, and the state and recipient markers that govern behavior are short and infrequent relative to response text, yet they carry the decisive information (Section 3.2). **(iii) Decisions compound.** Every emitted message joins the context of later steps, so restraint, non-repetition, and timeliness are properties of a trajectory and cannot be judged one chunk at a time (Section 3.3).

3 LIVE ASSISTANT

LIVEASSISTANT learns a streaming policy over synchronized livestream trajectories for the decision problem of Section 2. A data engine converts replay media and platform logs into chunk-aligned supervision. As shown in Figure 3, a role-conditioned Omni model (Xu et al., 2025) is optimized in two stages that address the second and third difficulties of Section 2.2. MA-MSFT establishes the structured action protocol, whereas SM-GSPO then corrects the action mismatch that arises once each prediction feeds back into the model's streaming context.

3.1 LIVESTREAM TRAJECTORY CONSTRUCTION

We reconstruct recorded livestreams and synchronized platform logs into 10-second multimodal intervals x_t, comprising native audio and video with aligned comments, gifts, viewer dynamics, and room metadata. At interval t, Gemini 3.1 Pro receives only $\{x_1, \ldots, x_t\}$ and prior teacher actions $\{y_1, \ldots, y_{t-1}\}$, and generates one structured OBS, MEM, or ANS action following Equation 5. Each request appends only the new interval while preserving the shared prefix, enabling implicit cache reuse. All trajectories undergo automatic validation, while benchmark trajectories additionally receive human review. Rooms are split before contiguous training windows are constructed. Full construction and serialization details are provided in Section C.1.

3.2 MARKER-AWARE MULTITURN SUPERVISED FINE-TUNING

The structured targets in Equation 5 are highly asymmetric in length: an OBS decision may contribute one supervised token, whereas MEM and ANS contain many content tokens, so token-averaged cross-entropy gives silence and state selection far less gradient than response wording. Let $y_{t,j}$ be the j^{th} assistant token of action y_t, let $\chi_{t,j} \in \{0,1\}$ indicate whether it is supervised, and let $\eta(y_{t,j})$ denote its marker-aware weight. We retain the full-vocabulary likelihood but add weight on protocol markers:

$$\mathcal{L}_{\text{vocab}} = -\frac{\sum_{t,j} \chi_{t,j}\eta(y_{t,j}) \log \pi_\theta(y_{t,j} \mid h_t)}{\sum_{t,j} \chi_{t,j}\eta(y_{t,j})}. \tag{6}$$

where $\eta(\cdot)$ distinguishes state, recipient, task, and vanilla content tokens, preserving open-ended generation while keeping rare structural decisions from being overwhelmed. Marker weighting still scores the state against the full vocabulary and does not shape the three-way boundary, so we form a restricted distribution $\pi_\theta^{\mathcal{S}}(z_t \mid h_t) = \text{softmax}_{z\in\mathcal{S}}\, \xi_t(z)$ from the same logits and apply a class-weighted focal objective

$$\mathcal{L}_{\text{state}} = -\frac{\sum_{t=1}^{T} \beta_{z_t}\, \text{sg}\big[(1-\pi_\theta^{\mathcal{S}}(z_t \mid h_t))^\gamma\big] \log \pi_\theta^{\mathcal{S}}(z_t \mid h_t)}{\sum_{t=1}^{T} \beta_{z_t}}. \tag{7}$$

where β_{z_t} compensates for state imbalance, γ emphasizes uncertain decisions (Lin et al., 2017), and sg stops the focal weight from becoming an optimization path. Restricted label-token objectives have been used for classification (Schick & Schütze, 2021); here the auxiliary distribution supervises only the state, while the full-vocabulary head is kept for generation. The complete objective is $\mathcal{L}_{\text{MA-MSFT}} = \mathcal{L}_{\text{vocab}} + \alpha\mathcal{L}_{\text{state}}$. Because training uses reference histories, MA-MSFT does not expose the policy to its own earlier errors, motivating the second stage.

3.3 STREAMING MULTITURN GSPO

MA-MSFT is trained with reference histories, whereas deployment conditions each decision on the model's previous actions. This exposure mismatch can propagate early activation or memory errors through a long stream (Bengio et al., 2015). SM-GSPO addresses it with self-generated trajectories and complementary turn- and trajectory-level credit, jointly optimizing state prediction, recipient routing, task selection, and response content.

On-policy multiturn rollout. At each training step, the resident rollout model is synchronized with the current training model and generates a group of G trajectories. Each turn conditions on the incoming stream and the model's preceding actions:

$$\widehat{y}_{g,t} \sim \mu\big(\cdot \mid x_{\le t}, \widehat{y}_{g,<t}\big), \tag{8}$$

where $g \in \{1,\dots,G\}$ indexes rollouts and μ is the behavior policy induced by the decoding parameters. The sampled trajectories remain fixed during optimization. Before the first update, the training model replays them to compute $\pi_{\theta_{\text{old}}}$, which is held fixed across subsequent optimizer steps.

Turn and trajectory rewards. Each turn receives a structural reward $\mathcal{R}^{\text{str}}_{g,t}$ for format, state, recipient, task, and message-count correctness. For a correctly predicted ANS state, a Qwen3-VL judge model provides a content score $c_{g,t} \in [0,1]$ (Section B.4.3). With a small state-shaping term $b_{g,t}$, the turn and trajectory rewards are

$$s_{g,t} = f_{g,t}\left(\mathcal{R}^{\text{str}}_{g,t} + \lambda_c c_{g,t} + b_{g,t}\right), \qquad R_g = \sum_{t=1}^{T_g} \mathcal{R}^{\text{str}}_{g,t}, \tag{9}$$

where $c_{g,t} = 0$ when content evaluation is not applicable. The turn reward provides local feedback, while the trajectory reward measures structural consistency over the complete stream.

Scoped credit assignment. The two rewards in Equation 9 carry complementary signals: $s_{g,t}$ scores each local decision, while R_g reflects restraint and consistency that emerge only across the full stream. We therefore route each reward to the tokens whose behavior it actually reflects (Figure 3):

$s_{g,t}$ to the tokens of turn t, and R_g to all assistant tokens of rollout g. Concretely, we standardize $\{s_{g,t}\}_{g=1}^{G}$ and $\{R_g\}_{g=1}^{G}$ within each rollout group to obtain the turn advantage $A_{g,t}$ and trajectory advantage A_g. Let $\mathcal{I}_{g,t}$ denote the assistant-token positions generated at turn t of rollout g. The turn advantage is assigned only to these tokens, with the GSPO ratio

$$\rho_{g,t} = \exp\left[\frac{1}{|\mathcal{I}_{g,t}|}\sum_{j\in\mathcal{I}_{g,t}} \log\frac{\pi_\theta(\widehat{y}_{g,t,j} \mid H_{g,t,j})}{\pi_{\theta_{\text{old}}}(\widehat{y}_{g,t,j} \mid H_{g,t,j})}\right], \tag{10}$$

where $H_{g,t,j} = (p, x_1, \widehat{y}_{g,1}, \ldots, x_{t-1}, \widehat{y}_{g,t-1}, x_t, \widehat{y}_{g,t,<j})$ is the complete prefix preceding token j. The trajectory advantage is assigned to all assistant tokens in rollout g, with the corresponding trajectory-level ratio ρ_g. Thus, turn-level credit optimizes each local decision, while trajectory-level credit captures its contribution to behavior over the complete stream.

Training objective. Using the standard clipped GSPO surrogate ℓ (Zheng et al., 2025), the final objective combines turn- and trajectory-level credit:

$$\mathcal{L}_{\text{SM-GSPO}} = \frac{1}{\sum_g T_g}\sum_{g,t}\ell(\rho_{g,t}, A_{g,t}) + \frac{1}{G}\sum_g \ell(\rho_g, A_g). \tag{11}$$

The first term trains each turn from its own reward, while the second promotes consistent behavior over the complete stream.

3.4 Long-Stream Inference

At inference, each chunk appends new media and platform tokens to a continuous streaming cache. The runtime keeps the first turn system instruction and recent 12 turns while older audio and visual states are evicted, and compact MEM and text states preserve semantic continuity, following the separation between dense recent perception and compressed long-term memory in streaming video systems (Xu et al., 2026; Liu et al., 2026). Details refer to Section B.6.

4 Experiments

We establish six research questions and conduct experiments to answer each of them. **(Q1)** Does the policy make the correct OBS, MEM, or ANS decision at each step? **(Q2)** Do native audio, video, comments, and gifts contribute complementary rather than redundant evidence? **(Q3)** Can one policy route useful feedback to viewers and hosts? **(Q4)** What do MA-MSFT and SM-GSPO help? **(Q5)** Does the semantic content written by MEM improve later decisions? **(Q6)** Do quality and efficiency survive long livestreams under a bounded context budget? Section 4.1 summarizes the setup; Section 4.2 addresses Q1, Q3, and Q4, while Section 4.3 isolates Q2, Q5, and Q6.

4.1 Experimental Setup

Training Data and Benchmark. We split streams by room before constructing model inputs, yielding 115 rooms for MA-MSFT, eight rooms for SM-GSPO, and 14 additional rooms for evaluation. Training samples are contiguous windows of 5–12 turns, while benchmark samples retain complete 12–60-turn clips, with each turn spanning 10 seconds. The benchmark contains 275 clips and 13,781 chunks. As summarized in Figure 5, it spans nine content categories, seven languages, eight countries, a long-tailed task taxonomy, all three decision states, both recipient roles, and frequent social signals. After benchmark construction, two human reviewers independently evaluate the trajectories under the prefix-only protocol. Human–benchmark agreement reaches 92% for state, 87% for recipient, 83% for task, and 91% for content. Further construction and composition details are provided in Appendix C, with implementation details in Section B.1.

Baselines. We report two complementary tracks. *Streaming video models*, including Dispider (Qian et al., 2025), LiveCC (Chen et al., 2025), StreamingVLM (Xu et al., 2026), LiveStarPro (Yang et al., 2026), MMDuet2 (Wang et al., 2025c), and MOSS-VL-Realtime (OpenMOSS Team, 2026), run with their released weights and native inference stacks without any adaptation;

Table 1: Main comparison on the LIVEASSISTANT benchmark. State reports overall OBS/MEM/ANS acc, followed by per-state recall. Recipient reports overall routing accuracy.

		State decision				Recipient routing			Action	Content	Overall
Method	**Input**	**All**	**OBS**	**MEM**	**ANS**	**All**	**Viewer**	**Host**	**Task**	**Sem.**	**Avg.**
TRIVIAL BASELINES											
RANDOM$_{\text{ref. prior}}$	–	36.31	48.00	26.00	26.00	75.65	86.00	14.00	34.95	–	48.97
Always <obs>	–	46.73	100.00	0.00	0.00	0.00	0.00	0.00	0.00	–	15.58
Always <ans><viewer>[narration]	–	26.35	0.00	0.00	100.00	85.58	100.00	0.00	55.08	–	55.67
PROPRIETARY MODEL											
Gemini 3.1 flash	A+V+C+G	47.23	68.26	6.11	49.30	38.79	40.03	31.32	26.71	68.00	37.58
GPT-5.4	A+V+C+G	27.90	6.87	7.73	87.48	71.16	**78.41**	25.33	47.06	60.05	48.71
STREAMING VIDEO MODELS											
Dispider (Qian et al., 2025)	V	46.51	**96.44**	0.00	0.00	–	–	–	–	–	–
LiveCC (Chen et al., 2025)	V	27.06	56.10	0.00	0.00	–	–	–	–	–	–
StreamingVLM (Xu et al., 2026)	V	2.17	4.23	0.00	0.00	–	–	–	–	–	–
LiveStarPro (Yang et al., 2026)	V	42.85	88.86	0.00	0.00	–	–	–	–	–	–
MMDuet2 (Wang et al., 2025c)	V	0.82	1.64	0.11	0.00	–	–	–	–	–	–
MOSS-VL-Realtime (OpenMOSS Team, 2026)	A+V	18.29	37.74	0.00	0.00	–	–	–	–	–	–
GENERAL (M)LLMS WITH UNIFIED STREAMING EVALUATION											
Qwen3-VL-4B (Qwen Team, 2025)	V+C+G	42.17	32.42	23.48	79.06	66.87	75.86	11.45	41.94	27.32	50.33
Qwen3-VL-8B (Qwen Team, 2025)	V+C+G	52.51	49.09	33.39	78.05	50.34	52.61	36.35	38.24	22.93	47.03
Qwen3.5-9B (Qwen Team, 2026)	V+C+G	43.22	25.71	28.69	90.44	68.33	74.43	30.72	46.43	31.47	52.66
Qwen3.5-27B (Qwen Team, 2026)	V+C+G	43.30	23.79	35.07	87.92	71.05	76.48	37.55	40.12	46.29	51.49
Qwen3-Omni-30B-A3B (Xu et al., 2025)	A+V+C+G	32.59	14.11	3.73	**96.12**	48.65	47.76	54.05	49.23	25.11	43.49
InternVL3.5-8B (Wang et al., 2025b)	V+C+G	58.05	78.50	27.00	51.02	29.21	29.30	28.66	27.34	28.09	38.20
InternVL3.5-14B (Wang et al., 2025b)	V+C+G	33.13	16.92	3.44	93.09	49.36	49.33	49.50	30.51	33.68	37.67
MiniCPM-V-4.5 (Yu et al., 2025)	A+V+C+G	49.27	80.95	3.61	35.94	19.59	20.72	12.63	11.20	19.25	26.69
Phi-4 Multimodal (Microsoft, 2025)	V+C+G	43.70	74.97	19.23	9.91	7.19	8.33	0.20	3.69	12.68	18.19
LIVEASSISTANT WITH THE QWEN3-OMNI THINKER											
LIVEASSISTANT ★	**A+V+C+G**	**71.14**	78.84	**48.20**	79.26	**72.67**	75.57	**56.71**	**58.41**	**81.78**	**67.41**

A/V/C/G denote native audio, video, comments, and gifts. "–" denotes an unevaluated metric due to no valid content. **Bold** and underline mark the best and second-best value in each column, computed over all non-trivial baselines and the proposed model.

because their training objectives differ from ours, their outputs do not reliably follow the action, recipient, and task protocol, and we report the corresponding metrics as "–". *General multimodal models* follow the protocol poorly under our multi-turn streaming inference, so we evaluate them in a single-round setting: at each 10-second chunk we concatenate the prior stream context into one prompt, keeping the visible content identical to what LIVEASSISTANT sees.

Metrics. State is the overall OBS/MEM/ANS accuracy, with OBS, MEM, and ANS reporting per-state recall. Recipient is overall routing accuracy, with Viewer and Host reporting per-role accuracy. Task measures end-to-end correctness on reference ANS turns, assigning zero credit to incorrect state decisions. Sem. is a coverage-aware Gemini score for semantic consistency, grounding, actionability, and concision. We report $\text{Avg.} = (\text{State} + \text{Recipient} + \text{Task})/3$ when all three components are available. Further details are provided in Appendix B.

4.2 MAIN COMPARISON EXPERIMENTS

Baseline comparison. LIVEASSISTANT achieves 67.41 Avg., outperforming the strongest general-model baseline by 14.75 points (Table 1). Per-state results reveal clear policy biases. Qwen3-Omni-30B-A3B attains 96.12 ANS recall but only 32.59 State, while most general models favor viewer-directed responses. Native streaming models are trained with different objectives, and their outputs do not reliably follow our action and recipient protocol. We therefore report their Recipient, Task, Sem., and Avg. metrics as "–".

Training stages. Table 2 shows consistent gains across the training stages. Ordinary multiturn SFT is the standard multiturn objective without marker weighting. MA-MSFT assigns higher weights to the special action tokens, substantially improving state prediction. Turn-level GSPO retains only turn-level rewards and credit assignment, yielding further gains in routing and task selection. The complete MA-MSFT+SM-GSPO objective combines turn-level and trajectory-level rewards and credit assignment, achieving the strongest overall performance and better MEM recall. These objectives are detailed in Section 3.2 and Equation 11.

Table 2: Controlled ablation of the progressive training strategy.

	State decision				Recipient routing			Action	Content	Overall
Training strategy	**Overall accuracy**	**Observation recall**	**Memory recall**	**Answer recall**	**Overall accuracy**	**Viewer accuracy**	**Host accuracy**	**Task accuracy**	**Semantic score**	**Average**
Ordinary multiturn SFT	69.36	87.71	35.99	68.64	63.54	65.77	49.70	51.12	74.32	61.34
MA-MSFT	70.05	87.06	36.50	72.00	66.42	69.09	49.90	52.55	79.90	63.01
MA-MSFT + turn-level GSPO	69.98	**89.79**	28.52	74.64	68.81	71.76	50.50	54.46	80.19	64.42
MA-MSFT + SM-GSPO ★	**71.14**	78.84	**48.20**	**79.26**	**72.67**	**75.57**	**56.71**	**58.41**	**81.78**	**67.41**

Table 3: Ablations of synchronized input signals and semantic memory.

	State decision				Recipient routing			Action	Overall	
Condition	**All**	**OBS**	**MEM**	**ANS**	**All**	**Viewer**	**Host**	**Task**	**Avg.**	**Δ Avg.**
A. SYNCHRONIZED INPUT SIGNALS										
Full synchronized input	71.14	78.84	48.20	79.26	72.67	75.57	56.71	58.41	67.41	0.00
– without Native audio	67.65	92.75	19.87	68.71	63.71	68.10	32.72	41.93	57.76	-9.65
– without Video	67.47	92.67	20.38	67.80	62.78	67.04	29.47	40.67	56.97	-10.44
– without Comments and gifts	67.69	91.10	23.82	68.13	63.52	67.79	30.18	41.59	57.60	-9.81
B. SEMANTIC MEM CONTROLS										
Full model	71.14	78.84	48.20	79.26	72.67	75.57	56.71	58.41	67.41	0.00
MEM payload removed	61.27	77.97	34.14	76.40	65.55	72.80	21.04	46.75	57.86	-9.62
MEM payload shuffled within room	66.87	74.97	46.91	71.80	59.11	64.76	24.45	39.29	55.09	-12.39
No-MEM	53.47	84.79	7.72	41.00	34.50	36.16	24.45	23.91	37.29	-30.19

4.3 ANALYSIS AND ABLATIONS

Contribution of synchronized stream signals. Panel A of Table 3 removes one input source at a time from the full context during inference (Q2). Removing audio, video, or social signals reduces Avg. by 9.65, 10.44, and 9.81 points, respectively, with video producing the largest drop. All three ablations sharply reduce MEM recall and host-routing accuracy while increasing OBS recall above 91%. These results show that the trained model relies on each modality: removing any one signal at inference time degrades memory formation, intervention timing, and recipient selection.

Utility of semantic memory. Semantic memory improves later decisions when it preserves relevant trajectory context (Q5). During inference, removing the memory payload while keeping all MEM markers and their positions reduces Average by 9.62 points. Replacing each payload with a length-matched memory from another time in the same room further increases the drop to 12.39 points, indicating that the benefit depends on temporally relevant content rather than additional context tokens. Removing MEM entirely during inference causes the largest drop of 30.19 points, with clear degradation in answer prediction, recipient routing, and task accuracy. Thus, the gain comes from semantically meaningful, temporally aligned memory content.

Effect of context length. A short interaction history provides most of the benefit, while longer contexts yield diminishing returns. We evaluate LIVEASSISTANT with the preceding 0, 2, 4, 6, 8, 10, or 12 interaction turns retained. As shown in Table 4, Overall rises sharply from 25.67 without history to 63.44 with only two turns, then improves gradually to 67.41 with twelve turns. The largest continued gains occur in state prediction, particularly MEM, whereas recipient routing largely stabilizes once a short history is available. The 12-turn full-media run matches the standard 12-turn setting. Because these runs contain only the hard-set evaluation, we report structural metrics and omit the semantic content score.

Performance over stream progression. LIVEASSISTANT remains stable through most of the stream, with degradation concentrated in the final segment (Q6). We divide each trajectory into five chronological segments and evaluate them with the same metrics as Table 1. As shown in Table 5, Average stays nearly unchanged from T1 to T4 (67.76–68.43), but drops in T5. The late

Table 4: Ablation over the number of preceding interaction turns retained in context.

	State decision				Recipient routing			Action	Overall
Context turns	Overall accuracy	Observation recall	Memory recall	Answer recall	Overall accuracy	Viewer accuracy	Host accuracy	Task accuracy	Average
0	42.96	62.06	25.02	25.36	21.30	23.16	10.02	12.75	25.67
2	64.68	72.08	36.55	79.12	71.72	75.24	49.90	53.92	63.44
4	67.64	75.12	42.40	79.01	72.06	75.34	51.70	55.30	65.00
6	68.99	76.94	44.50	78.70	71.89	74.66	56.71	56.42	65.76
8	69.80	77.80	46.12	78.62	72.20	74.98	55.11	57.51	66.50
10	70.45	78.55	47.13	78.73	71.95	74.89	53.91	57.65	66.68
12 (full context)	**71.14**	**78.84**	**48.20**	**79.26**	**72.67**	**75.57**	**56.71**	**58.41**	**67.41**

Table 5: Performance over the chronological progression of benchmark streams. Each trajectory is divided into five equal-duration segments, where T1 is the earliest and T5 is the latest.

		State decision				Recipient routing			Action	Content	Overall
Time segment	Turns	Overall accuracy	Observation recall	Memory recall	Answer recall	Overall accuracy	Viewer accuracy	Host accuracy	Task accuracy	Semantic score	Average
T1 (earliest)	2,566	69.38	68.48	**61.69**	79.31	74.66	76.37	59.09	**59.82**	76.31	67.96
T2	2,749	71.19	77.05	52.32	79.76	**74.73**	**77.52**	57.00	59.38	79.35	**68.43**
T3	2,752	**71.44**	81.06	44.63	80.06	73.17	75.13	**61.00**	**59.82**	83.46	68.14
T4	2,749	71.41	82.42	41.11	**80.62**	73.13	76.70	53.77	58.74	**84.60**	67.76
T5 (latest)	2,965	71.43	**83.33**	40.57	76.93	68.33	72.44	46.46	54.86	84.19	64.88
All	**13,781**	71.14	78.84	48.20	79.26	72.67	75.57	56.71	58.41	81.78	67.41

decline is mainly driven by weaker memory and host-directed decisions: memory recall falls from 61.69 to 40.57, Host Accuracy from 59.09 to 46.46, and Task Accuracy from 59.82 to 54.86. In contrast, observation recall increases from 68.48 to 83.33. This indicates that long-stream degradation arises primarily from preserving useful context and routing interventions, rather than from deciding when to keep observing.

Qualitative analysis. Appendix D presents multilingual trajectories across diverse livestream domains, illustrating how LIVEASSISTANT behaves over evolving stream context. Successful cases show how the model coordinates OBS, MEM, and ANS, selectively remains silent, preserves useful context through private memory, and produces grounded responses. These decisions are supported by both audiovisual content and synchronized platform signals, while recipient routing determines whether the resulting assistance should be delivered to viewers or surfaced to the host. Comparisons with general multimodal and native streaming models further highlight the importance of temporal decision structure: baseline models often respond to locally salient content without maintaining the same pattern of restraint, memory use, and recipient-aware intervention across turns. The failure cases reveal the remaining challenges, including cross-lingual grounding, repeated responses to persistent comments, incorrect gift attribution, and poorly timed host interruptions.

5 CONCLUSION

We introduced LIVEASSISTANT, a mixed-initiative framework for role-conditioned assistance in multi-party livestreams. Different from responding to an explicit request, the assistant infers emerging needs from synchronized media and social signals and jointly decides whether to act, when to act, whom to assist, and what to communicate. We realize this formulation with a unified OBS/MEM/ANS policy, trained on time-aligned livestream trajectories with marker-aware supervision and streaming reinforcement. On a 39-hour human-verified benchmark, LIVEASSISTANT substantially improves joint decision making and supports effective assistance to both viewers and hosts. These results shift streaming assistance from reactive response generation toward selective participation in continuously evolving live social environments.

USE OF LARGE LANGUAGE MODELS

We disclose two uses of generative AI required under the ICLR policy. *(i) Synthetic data.* As detailed in Section 3.1 and Section C.1, Gemini 3.1 Pro produces per-chunk OBS/MEM/ANS annotations under a prefix-only protocol, with deterministic validation and independent two-annotator review on the benchmark yielding 92%/87%/83%/91% agreement on state, recipient, task, and content (Section 4.1). *(ii) Automatic judges.* A Qwen3-VL judge scores response semantics for SM-GSPO reward shaping (Section 3.3), while Gemini serves as the content judge for benchmark evaluation (Section 4.1). Under recommended disclosure, an LLM was used only to improve writing readability. All research questions, formulation, model design, and experimental protocol were developed by the authors, who reviewed all AI-assisted content and take full responsibility for it.

ETHICS STATEMENT

This work uses recordings of real livestreams containing human speech, video, comments, and gifts, and studies assistance that directly affects hosts and viewers. All source material was obtained in accordance with the originating platform terms, and its research use was approved by an internal legal, privacy, and compliance review at the hosting institution prior to any annotation or training. Personally identifying information in comments and metadata was removed or anonymized before use. Human annotators reviewed benchmark trajectories with informed consent and fair compensation. Regarding dual-use risk, proactive assistance could be misused for intrusive messaging, manipulation, or surveillance, and routing errors could expose information to unintended recipients. The action space explicitly supports abstention and private updates via OBS and MEM, but any production deployment additionally requires explicit user consent, human oversight, and auditing of over-intervention and routing errors. The released benchmark and models are intended for research on socially aware streaming assistance, not for covert monitoring of individuals.

REPRODUCIBILITY STATEMENT

We provide sufficient detail to reproduce **LIVEASSISTANT** end to end. The problem setting and structured action protocol are formalized in Section 2 and Equation 5; the trajectory engine, room-level split, and corpus statistics are described in Section C.1 and Section C.2, with all annotation, policy, and judge prompts reproduced in Section B.4. The MA-MSFT and SM-GSPO objectives (Equation 6–Equation 11, Equation 12) and every optimizer, rollout, and reward hyperparameter used for the reported numbers are given in Section B.2, Section B.3, and Table 7. The streaming evaluation protocol, parser, and metrics are specified in Section B.6, and all results are produced on the room-held-out benchmark under this identical protocol. Upon acceptance we will release the anonymized benchmark, training and evaluation code, and the final checkpoint, subject to platform terms and the safeguards in the Ethics Statement.

REFERENCES

James F. Allen, Curry I. Guinn, and Eric Horvitz. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications*, 14(5):14–23, 1999. doi: 10.1109/5254.796083.

Shehreen Azad, Vibhav Vineet, and Yogesh Singh 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 (CVPR)*, 2026.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pp. 1171–1179, 2015.

Dan Bohus and Eric Horvitz. Models for multiparty engagement in open-world dialog. In *Proceedings of the SIGDIAL 2009 Conference*, pp. 225–234, London, UK, 2009. Association for Computational Linguistics. doi: 10.3115/1708376.1708409.

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 (CVPR)*, pp. 18407–18418, 2024. doi: 10.1109/CVPR52733.2024.01742.

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. LiveCC: Learning video LLM with streaming speech transcription at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29083–29095, 2025. doi: 10.1109/CVPR52734.2025.02708.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. A survey on proactive dialogue systems: Problems, methods, and prospects. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6583–6591, 2023a. doi: 10.24963/ijcai.2023/738.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10602–10621, Singapore, 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.711.

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 (ICCV)*, pp. 13448–13459, 2025. doi: 10.1109/ICCV51701.2025.01249.

James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):119–146, 2005. doi: 10.1145/1057237.1057243.

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. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 80–99. Springer, 2026. doi: 10.1007/978-3-032-37255-0_5.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

Ke Guo, Changle Qu, Jiayaqi Cheng, Xiao Zhang, Shijun Wang, Xiaoyu Zhang, Xueliang Wang, Le Zhang, Lantao Hu, and Jun Xu. KuaiLive-M3: A multi-modal, multi-domain, and multi-feedback dataset for live streaming recommendation. *arXiv preprint arXiv:2607.24862*, 2026.

Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 159–166. ACM, 1999. doi: 10.1145/302979.303030.

Eric Horvitz and Johnson Apacible. Learning and reasoning about interruption. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI)*, pp. 20–27. ACM, 2003. doi: 10.1145/958432.958440.

Stephanie Houde, Kristina Brimijoin, Michael Muller, Steven I. Ross, Dario Andres Silva Moran, Gabriel Enrique Gonzalez, Siya Kunde, Morgan A. Foreman, and Justin D. Weisz. Controlling AI agent participation in group conversations: A human-centered approach. In *Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI)*, pp. 390–408, 2025. doi: 10.1145/3708359.3712089.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3328–3338, 2025. doi: 10.1109/CVPR52734.2025.00316.

Koji Inoue, Divesh Lala, Mikey Elmers, Keiko Ochi, and Tatsuya Kawahara. An LLM benchmark for addressee recognition in multi-modal multi-party dialogue. In Maria Ines Torres, Yuki Matsuda, Zoraida Callejas, Arantza del Pozo, and Luis Fernando D'Haro (eds.), *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pp. 330–334, Bilbao, Spain, May 2025. Association for Computational Linguistics. ISBN 979-8-89176-248-0. URL https://aclanthology.org/2025.iwsds-1.36/.

Nataša Jovanović, Rieks op den Akker, and Anton Nijholt. Addressee identification in face-to-face meetings. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 169–176, Trento, Italy, 2006. Association for Computational Linguistics. URL https://aclanthology.org/E06-1022/.

Julien Lalanne, Raphael Bournet, and Yi Yu. LiveChat: Video comment generation from audio-visual multimodal contexts. *arXiv preprint arXiv:2311.12826*, 2023.

Ao Li, Zihan Xiao, Zihao Yue, Boshen Xu, Linli Yao, Jiaze Li, Pei Fu, Jianzhong Ju, Jian Luan, and Qin Jin. StreamPro: From reactive perception to proactive decision-making in streaming video. *arXiv preprint arXiv:2605.16381*, 2026.

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 *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12147–12151, 2026. doi: 10.1109/ICASSP55912.2026.11463959.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

Zikang Liu, Longteng Guo, Handong Li, Ru Zhen, Xingjian He, Ruyi Ji, Xiaoming Ren, Yanhao Zhang, Haonan Lu, and Jing Liu. Thinking in streaming video. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 21–38, 2026. doi: 10.1007/978-3-032-37447-9_2.

Xudong Lu, Xueying Li, Annan Wang, Yang Bo, Jinpeng Chen, Zengliang Li, Nianzu Yang, Rui Liu, Xue Yang, Jingwen Hou, and Hongsheng Li. OmniInteract: Benchmarking real-world streaming interaction for real-time omnimodal assistants. *arXiv preprint arXiv:2605.26485*, 2026.

Qianli Ma, Yaowei Zheng, Zhelun Shi, Zhongkai Zhao, Bin Jia, Ziyue Huang, Zhiqi Lin, Youjie Li, Jiacheng Yang, Yanghua Peng, Zhi Zhang, and Xin Liu. OmniScale: Scaling any modality model training with model-centric distributed recipe zoo. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 24271–24280, 2026. doi: 10.1609/aaai.v40i29.39607. Preprint circulated as "VeOmni".

Shuming Ma, Lei Cui, Damai Dai, Furu Wei, and Xu Sun. LiveBot: Generating live video comments based on visual and textual contexts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6810–6817, 2019. doi: 10.1609/aaai.v33i01.33016810.

Microsoft. Phi-4-Mini technical report: Compact yet powerful multimodal language models via mixture-of-LoRAs. *arXiv preprint arXiv:2503.01743*, 2025.

Junbo Niu, Yifei Li, 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 (CVPR)*, pp. 18902–18913, June 2025. doi: 10.1109/CVPR52734.2025.01761.

OpenMOSS Team. MOSS-VL technical report. *arXiv preprint arXiv:2608.15045*, 2026.

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 (CVPR)*, pp. 24045–24055, 2025. doi: 10.1109/CVPR52734.2025.02239.

Yiran Qiao, Xiang Ao, Jing Chen, Yang Liu, Qiwei Zhong, and Qing He. Deja vu in plots: Leveraging cross-session evidence with retrieval-augmented LLMs for live streaming risk assessment. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1496–1506, 2026. doi: 10.1145/3805712.3809737.

Changle Qu, Sunhao Dai, Ke Guo, Xiao Zhang, Liqin Zhao, Shijun Wang, Yannan Niu, Lantao Hu, Han Li, and Jun Xu. KuaiLive: A real-time interactive dataset for live streaming recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3356–3364, 2026. doi: 10.1145/3805712.3808587.

Qwen Team. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026. Qwen blog post.

Dongchuan Ran, Linyu Ou, Xueheng Li, Wenwen Tong, Chenxu Guo, Hewei Guo, Kaibing Wang, and Lewei Lu. EgoPro-Bench: Benchmarking personalized proactive interaction in egocentric video streams. *arXiv preprint arXiv:2605.07299*, 2026.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735, December 1974. doi: 10.2307/412243.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 255–269. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.20.

Tan Tang, Yanhong Wu, Yingcai Wu, Lingyun Yu, and Yuhong Li. VideoModerator: A risk-aware framework for multimodal video moderation in e-commerce. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):846–856, 2022. doi: 10.1109/TVCG.2021.3114781.

Xueyun Tian, Wei Li, Bingbing Xu, Heng Dong, Yuanzhuo Wang, and Huawei Shen. ROMA: Real-time omni-multimodal assistant with interactive streaming understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 23018–23039. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1153.

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 (NeurIPS)*, volume 38, pp. 146647–146674, 2025a. doi: 10.52202/085713-4406.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Xiaodong Wang, Langling Huang, Zhirong Wu, Xu Zhao, Teng Xu, Xuhong Xia, and Peixi Peng. LiViBench: An omnimodal benchmark for interactive livestream video understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(31):26517–26525, 2026. doi: 10.1609/aaai.v40i31.39859.

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. *arXiv preprint arXiv:2512.06810*, 2025c.

Yuxuan Wang, Yueqian Wang, Bo Chen, Tong Wu, Dongyan Zhao, and Zilong Zheng. OmniMMI: A comprehensive multi-modal interaction benchmark in streaming video contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18925–18935, 2025d. doi: 10.1109/CVPR52734.2025.01763.

Quan Wei, Siliang Zeng, Chenliang Li, Zhongruo Wang, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via fine-grained reward structure and credit assignment, 2026. URL https://arxiv.org/abs/2505.11821.

Ming Xie, Zizheng Huang, Xudong Tan, Chao Wang, Xiangyu Zeng, Wenxiao Wu, Tao Chen, Limin Wang, and Yanwei Fu. StreamOV: Streaming omni-video understanding via evidence-guided memory and response triggering. *arXiv preprint arXiv:2605.25621*, 2026.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 61463–61475, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6445dd88ebb9a6a3afa0b126ad87fe41-Paper-Conference.pdf.

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 (ICLR)*, 2025a.

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 (NeurIPS)*, volume 38, pp. 35334–35372, 2025b. doi: 10.52202/085713-1050.

Zhenyu Yang, Kairui Zhang, Bing Wang, Shengsheng Qian, and Changsheng Xu. LiveStarPro: Proactive streaming video understanding with hierarchical memory for long-horizon streams. *arXiv preprint arXiv:2606.17798*, 2026.

Shuchang Ye, Jinqiang Yu, Zhujun Xiao, Yajing Kong, Yist Y. Lin, Yang Ma, Jiaxi Liu, Xiaolei Xu, and Zheng Yu. From failure taxonomy to intervention: A diagnostic methodology for industry-scale AVLM in video and live-streaming platform moderation. *arXiv preprint arXiv:2606.30059*, 2026.

Wei Chee Yew, Hailun Xu, Sanjay Saha, Xiaotian Fan, Hiok Hian Ong, David Yuchen Wang, Kanchan Sarkar, Zhenheng Yang, and Danhui Guan. Dynamic content moderation in livestreams: Combining supervised classification with MLLM-boosted similarity matching. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 2528–2538. Association for Computing Machinery, 2026. doi: 10.1145/3770854.3783936.

Ruizhi Yu, Keyang Zhong, Peng Liu, Qi Wu, Haoran Zhang, Yanhao Zhang, Chen Chen, and Haonan Lu. Click-to-Ask: An AI live streaming assistant with offline copywriting and online interactive QA. In *Companion Proceedings of the ACM Web Conference 2026*, pp. 192–195, 2026. doi: 10.1145/3774905.3793134.

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, 2025. URL https://arxiv.org/abs/2509.18154.

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.

Ruixiang Zhao, Jie Yang, Zijie Xin, Tianyi Wang, Fengyun Rao, Jing Lyu, and Xirong Li. OmniPro: A comprehensive benchmark for omni-proactive streaming video understanding. *arXiv preprint arXiv:2605.18577*, 2026.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

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 *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 504–521. Springer, 2026. doi: 10.1007/978-3-032-37044-0_28.

APPENDIX ROADMAP

This appendix provides the positioning, reproducibility details, data construction pipeline, and qualitative analysis of LIVEASSISTANT. The contents are organized as follows.

Appendix A Related Work — POSITIONING

Reviews proactive streaming models, livestream systems, and mixed-initiative interaction. It positions LIVEASSISTANT along the social-stream input and mixed-initiative output axes summarized in Table 6.

Appendix B Training and Evaluation Details — REPRODUCIBILITY

Describes the implementation framework, MA-MSFT and SM-GSPO training procedures, rollout-policy correction, annotation and inference prompts, hyperparameters, and streaming evaluation protocol.

Appendix C LiveAssistant Data — DATA

Details the reconstruction of real livestreams, multimodal temporal alignment, prefix-constrained action annotation, quality control, room-level partitioning, and corpus composition reported in Figure 5.

Appendix D Case Study — QUALITATIVE ANALYSIS

Presents multilingual examples of LIVEASSISTANT behaviors and turn-by-turn comparisons with general multimodal and native streaming baselines across diverse livestream scenarios.

A RELATED WORK

A.1 PROACTIVE AND ONLINE STREAMING VIDEO UNDERSTANDING

Online video models must perceive and reason over content as it unfolds. Early systems established the core mechanics of this setting, introducing online frame processing, persistent visual memory, and interaction during an ongoing stream (Zhang et al., 2024; Chen et al., 2024). Benchmarks then sharpened what streaming understanding should measure, separating backward tracing, present understanding, and responses that must wait for future evidence (Huang et al., 2025; Niu et al., 2025; Yang et al., 2025a), and StreamingBench and OmniMMI extended evaluation to continuous multimodal input and proactive reasoning (Lin et al., 2026; Wang et al., 2025d).

A more recent line learns whether and when to respond rather than answering on demand. Dispider separates perception, decision, and reaction; StreamBridge adds a learned activation mechanism; LiveStar learns when to respond and when to stay silent; and StreamReady judges whether the accumulated evidence is sufficient at the moment of emission (Qian et al., 2025; Wang et al., 2025a; Yang et al., 2025b; Azad et al., 2026). MMDuet2 and StreamPro optimize response timing and restraint with multiturn reinforcement learning (Wang et al., 2025c; Li et al., 2026), in line with broader efforts to design turn-level rewards that assign credit to individual decisions inside a multiturn interaction (Wei et al., 2026). A parallel effort targets native audio-visual modeling under bounded latency: LiveCC interleaves visual observation with streaming speech transcription, while ROMA and StreamOV consume native audio-visual input (Chen et al., 2025; Tian et al., 2026; Xie et al., 2026); LiveVLM, StreamingVLM, and LiveStarPro sustain stable memory and latency over long streams (Xu et al., 2026; Yang et al., 2026); and OmniPro and OmniInteract contribute evaluation protocols for audio-dependent proactive responses (Zhao et al., 2026; Lu et al., 2026). Throughout this line, proactivity remains a reaction to an already implied query or a salient scene event, the response is addressed to a single user, the need is never discovered from the environment, and no recipient is selected among several parties.

A.2 LIVESTREAM AS A SOCIAL AND PLATFORM ENVIRONMENT

A separate body of work models the livestream medium itself, yet it treats comment and platform signals through four narrow lenses. As generation targets, LiveBot produces live comments from visual and textual context, and LiveChat produces comments from audio-visual context together with conversational history (Ma et al., 2019; Lalanne et al., 2023). As retrieval evidence, LiViBench

Table 6: **Comparison of prior work along the two axes of livestream assistance.** *Social stream input*: native audio and video (**A/V**; ● both, ⊙ video only), endogenous requests discovered from the comment stream (**Req.**), and platform signals such as gifts, viewer dynamics, and room metadata (**Plat.**). *Mixed-initiative output*: whether, when, whom, and what. **Whom** counts distinct recipient roles, and ○ marks systems that never address a role. No prior line jointly discovers endogenous requests, decides when to act, and routes grounded feedback between viewers and hosts. Support is full (●), partial (⊙), or none (○). **Type**: Benchmark (B), Model (M), System (S). **Dom.**: general video (V), livestream (L). † concurrent work, ⋆ ours.

Work	Type	Dom.	Social stream input			Mixed-initiative output			
			A/V	Req.	Plat.	Whether	When	Whom	What
StreamingBench (Lin et al., 2026)	B	V	●	○	○	⊙	⊙	1	●
OVO-Bench (Niu et al., 2025)	B	V	⊙	○	○	○	⊙	1	●
OmniMMI (Wang et al., 2025d)	B+M	V	●	○	○	●	●	1	●
StreamMind (Ding et al., 2025)	M	V	●	○	○	●	●	1	●
LiveStar (Yang et al., 2025b)	M	V	⊙	○	○	●	●	1	●
MMDuet2 (Wang et al., 2025c)	M	V	⊙	○	○	●	●	1	●
StreamPro (Li et al., 2026)	M	V	⊙	○	○	●	●	1	●
ROMA (Tian et al., 2026)	M	V	●	○	○	●	●	1	●
StreamOV (Xie et al., 2026)	M	V	●	○	○	●	●	1	●
VST† (Guan et al., 2026)	M	V	⊙	○	○	●	●	1	●
Em-Garde† (Zheng et al., 2026)	M	V	●	○	○	●	●	1	●
EgoPro-Bench† (Ran et al., 2026)	B	V	●	○	○	⊙	⊙	1	●
LiViBench (Wang et al., 2026)	B	L	●	⊙	⊙	○	○	–	●
Click-to-Ask (Yu et al., 2026)	S	L	⊙	⊙	⊙	⊙	⊙	1	●
VideoModerator (Tang et al., 2022)	S	L	●	○	⊙	⊙	○	1	⊙
LIVEASSISTANT ⋆	B+M+S	L	●	●	●	●	●	2	●

brings native audio, speech, and real-time comments into livestream video question answering, but the query is still supplied externally and comments serve only as supporting evidence (Wang et al., 2026). As recommendation features, KuaiLive and KuaiLive M3 expose platform interactions, behavioral feedback, and temporally evolving content representations, yet they target recommendation and highlight prediction rather than online stream understanding (Qu et al., 2026; Guo et al., 2026). As risk evidence, livestream moderation is cast as a dedicated detection pipeline: VideoModerator couples audio-visual risk modeling with a human review interface for commerce streams (Tang et al., 2022), and recent industrial systems add multimodal similarity matching, intervention-oriented failure taxonomies, and cross-session evidence retrieval (Yew et al., 2026; Ye et al., 2026; Qiao et al., 2026). The system closest to ours, Click-to-Ask, answers viewer inquiries with streaming memory, but the host manually selects each question and feedback returns to a single party, so request discovery and routing are again supplied from outside (Yu et al., 2026). In every case the system commits in advance to one behavior, whether always generating, always answering, always recommending, or always scoring, and never arbitrates among staying silent, answering a viewer, or cueing the host.

A.3 Mixed Initiative and Proactive Interaction

Deciding whether to act, when to intervene, and who should hold the initiative is a long-standing question in interactive systems. Mixed-initiative interfaces frame assistance as an expected-utility tradeoff between automated action and user control, so that a system acts only when the expected benefit outweighs the cost of intrusion (Horvitz, 1999; Allen et al., 1999). A related line studies when an interruption is appropriate and treats staying silent as a first-class decision rather than a failure to respond (Horvitz & Apacible, 2003; Fogarty et al., 2005). Once more than one interlocutor is present, when and to whom to speak become explicit choices: conversation analysis characterizes turn-taking as a jointly managed allocation of speakership (Sacks et al., 1974), addressee detection identifies the intended recipient of an utterance in multi-party meetings (Jovanović et al., 2006), and situated agents model engagement to decide when to enter or hold a multi-party conversation (Bohus & Horvitz, 2009). Recent work carries these ideas into language agents, where

proactive dialogue systems learn to ask clarifying questions, steer toward a goal, and judge when to take the initiative (Deng et al., 2023b;a). Together these works supply the decision vocabulary that livestream assistance requires, namely whether, when, and toward whom to act. They are nonetheless developed in symbolic, single-modality, or face-to-face settings with a small and known set of interlocutors, and none instantiates this vocabulary over a native audio-visual stream in which the need must first be discovered from a noisy crowd.

A.4 Positioning

Table 6 places these three lines on the two axes that define livestream assistance, the social stream input a system consumes and the mixed-initiative output it produces, and each line advances one axis while leaving the coupling open. Proactive streaming models decide whether and when to speak, but they address a single recipient and never discover the need, so they fill the output axis only for one role and stay empty on endogenous requests. Livestream systems ingest native media together with platform signals, yet each fixes one behavior rather than arbitrating among actions, so they gain partial input coverage while remaining sparse across the output axis. Mixed-initiative interaction contributes the whether, when, and whom vocabulary, but it has not been instantiated over a native audio-visual social stream and therefore does not appear as a system in the table.

LiveAssistant studies the coupling that these lines leave open. Its unit of prediction is a single structured action over synchronized media and platform signals, in which silence, private memory, recipient selection, task selection, and grounded language are produced by one streaming policy. As the final row of Table 6 shows, this is the only entry that is full on both axes, discovering endogenous requests from the comment stream, consuming native audio, video, and platform signals, and routing grounded feedback across two recipient roles. Several systems in the table are concurrent with our work, including ROMA, StreamOV, OmniPro, OmniInteract, StreamPro, LiViBench, and Click-to-Ask; to the best of our knowledge, none jointly performs endogenous request discovery, timing, recipient routing, and role-conditioned grounding within a single policy.

B Training and Evaluation Details

This section reports the implementation, prompts, optimization settings, and evaluation protocol used for **LiveAssistant**.

B.1 Implementation Framework

Base model. We initialize **LiveAssistant** from the thinker component of Qwen3-Omni (Xu et al., 2025) and add the action protocol tokens to its tokenizer. Native audio and video are processed by the corresponding modality encoders. Comments, gifts, highlight indicators, and room metadata are serialized as timestamped text. These inputs and previous assistant actions are arranged chronologically according to Equation 5. We optimize only the text generating thinker policy and exclude speech generation from both training stages.

Training stack. MA-MSFT is implemented with veOmni (Ma et al., 2026) and FSDP2 (Gugger et al., 2022). SM-GSPO uses a custom distributed framework that separates streaming rollout generation from policy optimization. The rollout model generates trajectories incrementally with KV cache reuse, while the training workers replay the sampled trajectories to compute the old and current policy log probabilities.

B.2 MA-MSFT Training Details

MA-MSFT is trained on contiguous multiturn windows from the room level training split described in Section C.1. Each example contains synchronized media chunks, platform signals, and a reference OBS, MEM, or ANS action at every interval. Loss is applied only to assistant outputs. The objective combines full vocabulary language modeling, marker aware token weighting, and the auxiliary state objective defined in Section 3.2. The resulting checkpoint initializes SM-GSPO. Complete settings are reported in Table 7.

B.3 SM-GSPO Training Details

SM-GSPO starts from the MA-MSFT checkpoint. Before each rollout step, the resident rollout model is synchronized with the training model and generates trajectories conditioned on its own preceding actions. The sampled tokens remain fixed during optimization. Before the first update, the training model replays each trajectory to compute the anchor log probabilities $\log \pi_{\theta_{\text{old}}}$. These values remain frozen while $\log \pi_\theta$ is recomputed at every optimizer update. Rank 0 hosts the resident rollout model, and the remaining seven ranks perform policy optimization.

Rollout policy correction. Decoding transforms the synchronized rollout model into a behavior policy μ that can differ from the recomputed anchor $\pi_{\theta_{\text{old}}}$. We correct this mismatch using truncated prefix importance weighting. Let $\mathcal{I}_{g,\tau}$ contain the assistant tokens generated at turn τ of rollout g. The correction for turn t is

$$\omega_{g,t} = \min\left[\bar{\omega}, \exp\left(\sum_{\tau=1}^{t}\sum_{j\in\mathcal{I}_{g,\tau}} \log \frac{\pi_{\theta_{\text{old}}}(\widehat{y}_{g,\tau,j} \mid H_{g,\tau,j})}{\mu(\widehat{y}_{g,\tau,j} \mid H_{g,\tau,j})}\right)\right], \qquad (12)$$
$$\widetilde{A}_{g,t} = \omega_{g,t} A_{g,t}.$$

The prefix determines only the correction weight. The advantage $\widetilde{A}_{g,t}$ remains assigned exclusively to the assistant tokens generated at turn t. The trajectory correction is defined analogously over the complete rollout, yielding $\widetilde{A}_g = \omega_g A_g$. The implemented objective uses these corrected advantages in Equation 11.

B.4 Prompts and Automated Supervision

We use different prompts for trajectory annotation, policy inference, and semantic content evaluation. The annotation and policy prompts share the OBS, MEM, and ANS protocol. The content judge evaluates only the semantic quality of matched external responses.

B.4.1 Trajectory Annotation Prompt

Trajectory annotation

Role
Convert one continuous livestream clip into chunk level actions for an online omni modal assistant.

Input
- One clip between 2 and 10 minutes with native audio
- Consecutive decision intervals of 10 seconds
- Time aligned comment and gift summaries
- Room metadata and optional carryover context

Action protocol
For every chunk t, emit exactly one state:
- OBS: remain silent
- MEM: store a short private semantic memory
- ANS: send one or more messages to a viewer or host

Evidence constraint
The decision for chunk t may use only information observed through that chunk. Every MEM or ANS action must provide supporting indices $E_t \subseteq \{1, \ldots, t\}$. Future events must not influence the action.

Task taxonomy
- Viewer tasks: `narration`, `key_event`, `viewer_qa`, `entity_pinning`, `newcomer_recap`
- Host tasks: `comment_question_surge`, `faq_gap`, `pacing_alert`, `highlight_moment`

Output fields

chunk_id	current chunk index
state	OBS, MEM, or ANS
mem	private memory and evidence chunks
audience	viewer or host
task	one task from the taxonomy
text	concise grounded message
evidence_chunks	supporting chunk indices
confidence	low, medium, or high

Prefer OBS when evidence is insufficient. Avoid repeated or low value interventions. Each external message must be grounded, limited to one sentence, and no longer than 30 words.

B.4.2 Policy and Inference Prompt

Streaming policy instruction

Role
Observe a livestream one chunk at a time and decide whether assistance is needed. Use native video, audio, comments, gifts, highlight indicators, room metadata, and the retained interaction history.

Output protocol
Emit exactly one of the following forms:

```
<obs>
<mem> private semantic memory </mem>
<ans><viewer>[task] concise message </ans>
<ans><host>[task] concise message </ans>
```

Constraints
- Emit the state marker before memory or message content
- Use one valid task tag for every external message
- Remain silent when no useful intervention is needed
- Do not repeat information already communicated
- Keep every message grounded and under 30 words

Per turn input

chunk	identifier and time interval
media	current video with native audio
comments	count and repeated comments
gifts	count, total value, and maximum value
highlight	interval highlight indicator

B.4.3 Content Judge Prompt

Semantic content judge

Role
Evaluate whether the predicted message conveys the same useful meaning as the reference.

Instructions
- Judge semantic equivalence rather than lexical overlap
- Ignore state markers, recipient markers, task tags, and formatting
- Penalize missing information, contradictions, hallucinations, and irrelevant content
- Return one scalar score between 0.0 and 1.0

Rubric

1.0	Equivalent useful content
0.7	Mostly correct with minor omissions
0.4	Partially relevant but missing important information
0.0	Contradictory, hallucinated, irrelevant, or empty

Input and output

Table 7: Training configurations for MA-MSFT and SM-GSPO. Both stages update all model parameters.

MA-MSFT		SM-GSPO	
Configuration	**Value**	**Configuration**	**Value**
Initialization	Qwen3-Omni thinker with action tokens	Initialization	MA-MSFT checkpoint
Framework	veOmni with FSDP2	Framework	Custom rollout framework with ZeRO 3
Training budget	3 epochs Global batch 512; micro batch 1;	Training budget	1328 rollout steps 2 prompts per step; group size 5;
Batch construction	gradient accumulation 64	Rollout batch	10 trajectories per step
Optimizer	AdamW	Updates per rollout	3
Learning rate	1×10^{-5} Cosine; warmup ratio 0.007;	Learning rate	1×10^{-7} Cosine; 10 warmup steps;
Schedule	minimum LR 1×10^{-7}	Schedule	minimum LR ratio 0.1
Weight decay	0	Gradient clipping	1.0
Sequence length	32,768 tokens	Rollout horizon	12 turns
Generation	Not applicable	Generation	32 new tokens per turn
Precision	bfloat16 with FP32 reduction 8 GPUs; data shard 8; expert parallel 8;	Precision	bfloat16 1 resident rollout GPU;
Parallelism	tensor, pipeline, and context parallel 1 1 fps; 2 to 10 frames;	Parallelism	7 ZeRO 3 training GPUs Temperature 0.5; top p 1.0;
Video sampling	maximum 100,352 pixels	Decoding	top k 0; repetition penalty 1.0
Audio	16 kHz; retained with video Token weights 2.0 to 8.0;	Group size G	5
Marker objective	state loss weight 0.1	GSPO clipping	Low 3×10^{-4}; high 4×10^{-4}
State weights	OBS/MEM/ANS = 3/1/1 FlashAttention 2; fused Triton MoE;	Correction cap $\bar{\omega}$	2.0
Kernels	Liger cross entropy	Hard reward weights	State/task/recipient/count = 3/3/1/1 State match $+0.35$;
Gradient norm	1.0	Reward shaping	OBS miss -0.35; MEM miss -0.60

```
REFERENCE   reference message
PREDICTED   predicted message
SCORE       scalar from 0.0 to 1.0
```

B.5 Training Hyperparameters

Table 7 reports the optimization, batching, parallelism, media processing, rollout, and reward settings for both training stages.

B.6 Evaluation Protocol

Streaming inference. Each benchmark stream is processed chronologically in intervals of 10 seconds. At turn t, the model receives the current media chunk, synchronized platform signals, and retained interaction history. Evaluation uses the inference prompt as given above, greedy decoding, and a maximum generation length of 64 tokens per turn. Inference retains an active multimodal window of 120 seconds, 12 recent text turns, and a target KV cache budget of 32,768 tokens. The model achieves an average end-to-end turn latency of 3.996 s/turn/GPU, corresponding to a throughput of 0.250 turns/s/GPU

Structured parsing and metrics. The evaluation uses a rule-based parser following Equation 5. It requires one complete `<obs>`, `<mem>...</mem>`, or `<ans>...</ans>` action. Each answer must include a valid recipient marker and task tag. State prediction is evaluated on all turns. Recipient routing and task selection are evaluated when the reference state is ANS. Semantic evaluation is optional and is applied only after the structured fields have been scored. It does not affect the main hard rule results. All metrics follow Section 4.1.

Held out benchmark. Evaluation uses the room held out split described in Section C.1. No room, clip, or decision interval is shared with the optimization data, and benchmark examples preserve complete clips for long stream evaluation.

C Details About LiveAssistant Data

C.1 Constructing Streaming Assistant Data from Real Livestreams

We transform offline recorded livestreams and synchronized platform logs into complete trajectories that preserve the information available at each online decision point. Each trajectory combines native video and audio, aggregated platform activity, and structured assistant actions. The pipeline consists of stream reconstruction, chunk-level alignment, prefix-constrained annotation, action serialization, quality control, and room-level partitioning.

Stream reconstruction. The raw media consist of timestamped HLS transport-stream slices collected from real livestream sessions, together with platform logs such as comments, gifts, and room metadata. We recover continuous viewing intervals by ordering the slices according to their wall-clock timestamps and stitching adjacent slices. A new clip begins whenever the gap between two consecutive slices exceeds two seconds, preventing disconnected replay fragments from being treated as one continuous interaction. We retain clips between 2 and 10 minutes and preserve the original timestamps of all platform events for subsequent alignment.

Chunk-level multimodal alignment. Each reconstructed clip is divided into decision intervals of $\Delta = 10$ seconds. For interval t, the pipeline constructs

$$x_t = (v_t, a_t, e_t), \tag{13}$$

where v_t and a_t are the native video and audio within the interval, and e_t contains the synchronized platform signals. These signals include aggregated comment activity, representative viewer requests, gift activity, and room metadata.

Prefix-incremental action annotation. We query Gemini 3.1 Pro once per decision interval using only the stream prefix available at that time. At interval t, the teacher receives

$$c_t = (p, x_1, y_1, \ldots, x_{t-1}, y_{t-1}, x_t), \tag{14}$$

where p is the persistent annotation instruction and $y_{<t}$ are the teacher's preceding actions. The full clip and all intervals after t are never included in the request. Each response therefore produces exactly one decision from information available online, eliminating future-information access by construction.

To avoid repeatedly processing the growing prefix, each 10-second multimodal interval is stored as an immutable input unit and appended in temporal order. Consecutive API requests preserve the identical system instruction and previously submitted units, allowing Gemini's implicit prefix cache to reuse the shared context when available; we do not re-encode each prefix as a newly concatenated video.

For each interval, the teacher predicts

$$z_t \in \{\text{OBS}, \text{MEM}, \text{ANS}\}, \tag{15}$$

together with the fields required by the selected state. OBS represents silence. MEM records an internal semantic update u_t without producing an external message. ANS contains one or more external messages, each specifying a recipient, task type, and open-vocabulary response. The output also records an evidence set

$$E_t \subseteq \{1, \ldots, t\}, \tag{16}$$

which is checked during validation.

The task vocabulary (in Figure 5) covers viewer narration, key-event explanation, question answering, entity pinning, and newcomer recaps, as well as host-facing support for question surges, unresolved frequently asked questions, pacing issues, and highlight moments. These labels distinguish whether the assistant should answer viewers directly or route useful information to the host.

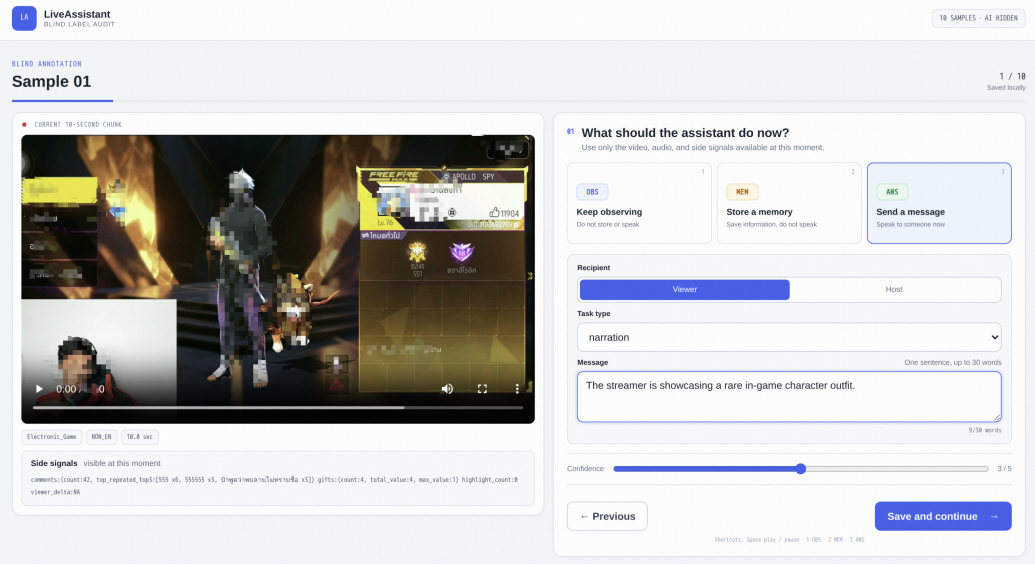


Figure 4: Blind, independent annotation interface. Reviewers walk each sampled trajectory chunk by chunk from the first chunk under the same prefix-only protocol as the benchmark, seeing the current video, audio and social signals plus preceding chunks and *only their own* earlier decisions; future chunks, the other reviewer's decisions, Gemini 3.1 Pro benchmark labels, and any model predictions are hidden throughout, so state (OBS/MEM/ANS), recipient, task type, short message, and 1–5 confidence are elicited without anchoring.

Action serialization. Each teacher decision is converted into the structured target y_t defined in Equation 5. The first target token specifies the state. For MEM, the state marker is followed by the semantic memory. For ANS, recipient and task markers precede the response content. Annotation-only fields, including evidence indices, confidence scores, triggers, and validation flags, are removed from the supervised target. The resulting trajectory is

$$(p, x_1, y_1, \ldots, x_T, y_T), \tag{17}$$

where p is the persistent system instruction. Previous assistant actions remain in the history because later decisions at deployment condition on the model's own preceding outputs.

Model representation. The policy uses the thinker component of Qwen3-Omni (Xu et al., 2025). Media tokens, platform signals, and previous assistant actions are arranged in temporal order. The resulting sequence allows one autoregressive policy to determine whether and when to act, whether to update memory, whom to address, which task to perform, and what content to produce.

Automatic quality control. Every generated trajectory passes deterministic validation and normalization. The validator checks schema validity, state-payload consistency, recipient-task compatibility, output formatting, duplicated actions, and evidence ordering. In particular, every evidence index in E_t must be no greater than t. It also enforces that OBS contains neither memory nor an external response, that MEM contains no external message, and that every ANS action contains the required recipient and task markers. Invalid fields are repaired when the correction is unambiguous; otherwise, the affected example is removed.

Human evaluation. To externally check the Gemini 3.1 Pro labels, we have two reviewers *independently re-annotate them from scratch* under the blind protocol of Figure 4: each sees only past chunks and their own prior decisions, and never sees the Gemini labels, the other reviewer's labels, or any model prediction. Only *after* annotation do we compute (i) *human vs. withheld Gemini* accuracy, averaged over the two reviewers, and (ii) *inter-annotator* Cohen's κ on the categorical fields (Table 8); free-form content is scored by an LLM judge on a 1–5 Likert scale, and we also report the fraction of decisions on which the two reviewers agree within one point. Because reviewers are

Table 8: Blind human evaluation on benchmark trajectories, with Gemini 3.1 Pro labels withheld from annotators throughout (Figure 4). For categorical fields we report accuracy against the withheld labels and Cohen's κ between the two reviewers; for free-form content we report the LLM-judged 1–5 Likert mean and pairwise within-1 agreement.

Dimension	Human vs. withheld Gemini	Inter-annotator
State (OBS/MEM/ANS)	92.0% acc.	$\kappa = 0.86$
Recipient (viewer/host)	87.0% acc.	$\kappa = 0.81$
Task type	83.0% acc.	$\kappa = 0.74$
Content (LLM-judged Likert 1–5)	4.32 mean	91.0% within-1

blind to the benchmark labels throughout, the resulting agreement is an external check rather than a post-hoc audit, and the substantial-to-almost-perfect κ values indicate that the underlying labelling policy is reproducible by independent annotators.

Room-level splitting and window construction. We split streams by room before constructing training or evaluation examples. The final partition contains 115 rooms for MA-MSFT, 8 additional rooms for SM-GSPO, and 14 rooms for the benchmark. Thus, the two optimization stages use 123 rooms in total, with no room, clip, or decision interval shared across the MA-MSFT, SM-GSPO, and benchmark partitions. Each training example is a contiguous multiturn window from one stream, and intervals from unrelated rooms are never packed into the same sequence. Benchmark examples retain complete clips for chronological evaluation.

The combined optimization partition contains 115,965 decision intervals, while the benchmark contains 13,781. Their state distributions are similar: the optimization partition contains 45.90% OBS, 26.46% MEM, and 27.64% ANS, compared with 48.21%, 25.84%, and 25.95% in the benchmark. This room-level partitioning preserves the overall action composition while preventing room-specific leakage. Details can be seen in Section C.2.

C.2 Dataset Composition and Coverage

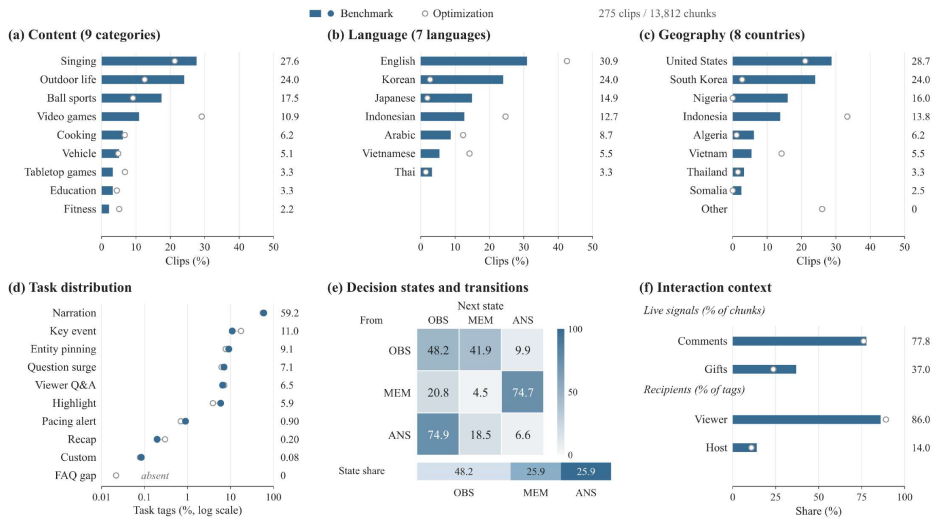


Figure 5: Statistics of LIVEASSISTANT. The training corpus combines the MA-MSFT and SM-GSPO partition. Unique chunk statistics remove overlap among multiturn windows.

Figure 5 summarizes the benchmark after deduplicating chunks shared by overlapping multiturn windows. Bars report benchmark composition, while hollow markers show the combined MA-MSFT and SM-GSPO optimization partition for comparison. The benchmark contains 275 complete clips and 13,781 unique 10-second chunks.

Content, language, and geography. As shown in Figure 5(a), the benchmark covers nine content categories. Singing accounts for 27.6% of clips, followed by outdoor life at 24.0%, ball sports at 17.5%, and video games at 10.9%. The remaining clips cover cooking, vehicles, tabletop games, education, and fitness. The language distribution in panel (b) spans English, Korean, Japanese, Indonesian, Arabic, Vietnamese, and Thai. English is the largest group at 30.9%, followed by Korean at 24.0%. Panel (c) includes eight named countries and an additional geographic tail, with the United States, South Korea, Nigeria, and Indonesia forming the largest groups. The optimization markers indicate broader coverage in several content, language, and geographic groups rather than an exact replication of the benchmark distribution.

Task distribution. External actions follow a long-tailed task distribution, as shown in Figure 5(d). Narration is the most frequent task at 59.2%, followed by key-event explanation at 11.0%, entity pinning at 9.1%, question-surge reporting at 7.1%, viewer question answering at 6.5%, and highlight detection at 5.9%. Pacing alerts, newcomer recaps, custom actions, and FAQ-gap detection form the sparse tail. This distribution reflects the prevalence of viewer-facing interpretation while retaining less frequent host-facing interventions.

Decision states and transitions. Panel (e) shows a balanced mixture of deliberate silence and active behavior: 48.2% of benchmark chunks are labeled OBS, while MEM and ANS each account for 25.9%. The transition matrix further reveals temporal structure in the annotations. An OBS action is followed by MEM in 41.9% of transitions, MEM proceeds to ANS in 74.7%, and ANS returns to OBS in 74.9%. These patterns capture a common progression in which the assistant first observes an event, retains relevant context, and responds only when an external action becomes useful.

Social context and recipients. Comments occur in 77.8% of benchmark chunks and gifts in 37.0%, providing time-aligned evidence about audience activity and socially salient moments. Among external messages, 86.0% are directed to viewers and 14.0% to hosts. Viewer-facing actions primarily provide narration, explanations, and direct answers, whereas host-facing actions surface audience demand, pacing issues, and notable moments. The corresponding optimization statistics in panel (f) exhibit the same overall pattern, with comments as the dominant platform signal and viewers as the primary recipient.

D Case Study

D.1 Diverse Behaviors of LiveAssistant

Figures 6–9 present four clips covering different languages, regions, and livestream domains. Each case contains four consecutive decision turns and illustrates how LiveAssistant coordinates observation, memory, response timing, and recipient selection as the stream evolves.

The cooking case follows an OBS → OBS → MEM → ANS trajectory, accumulating repeated audience requests before notifying the host. In the education case, the model tracks two consecutive stages of a mathematics lesson and reports the evolving audience consensus. The outdoor case demonstrates visually grounded assistance by retaining a viewer question and answering once the rice fields become identifiable. The singing case combines action selection with recipient routing by narrating the performance for viewers, recording a subsequent gift surge, and alerting the host. Together, these cases demonstrate multilingual temporal understanding, semantic memory, multimodal grounding, and adaptive routing within a unified streaming policy.

D.2 Qualitative Case Studies

We examine six four-turn cases that preserve consecutive 10-second media inputs, native audio, platform signals, and model-specific output histories. Three cases compare LiveAssistant with representative four general multimodal models, and three compare it with six native streaming baselines from Table 1. General models follow our structured action instruction, whereas streaming models retain their original output protocols.

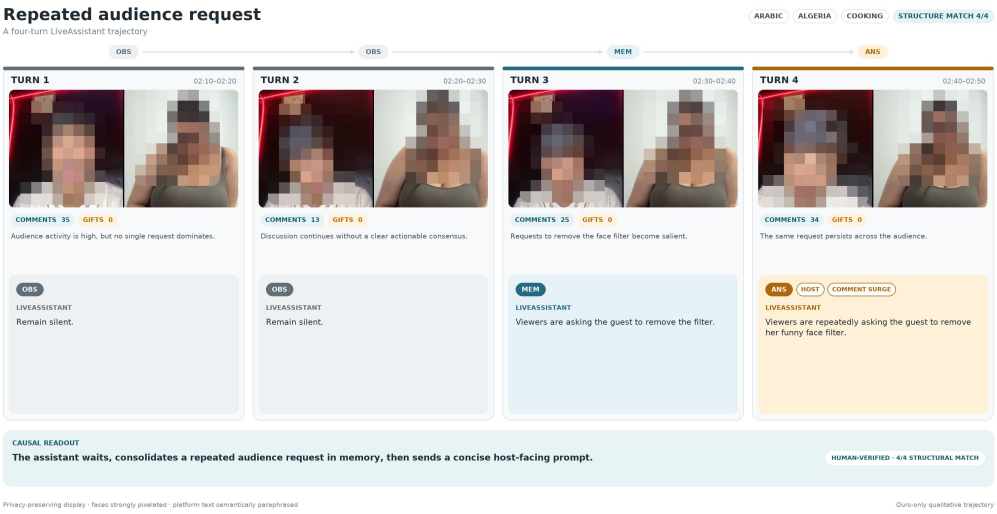


Figure 6: Arabic cooking. Persistent audience requests are consolidated into a host-facing notification.

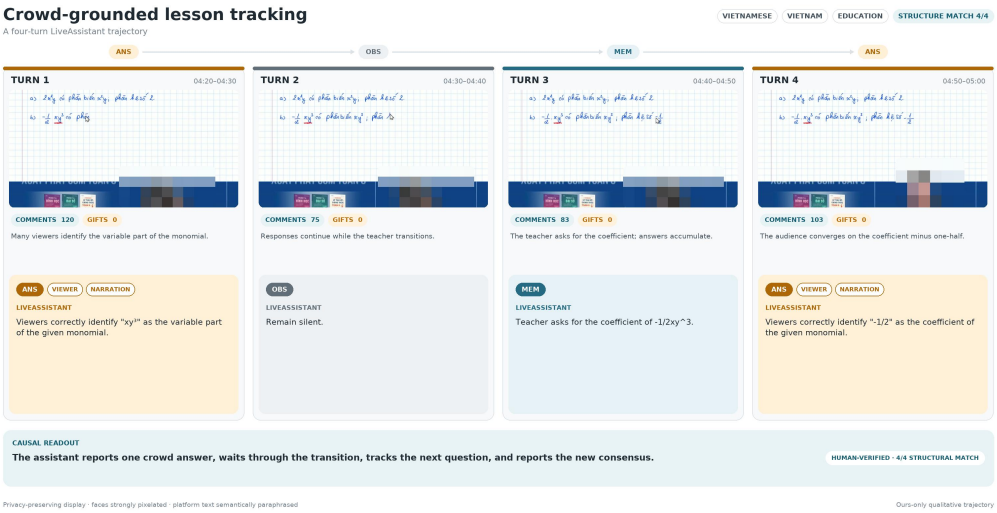


Figure 7: Vietnamese education. LIVEASSISTANT follows two successive questions and reports the corresponding answers.

General multimodal models. Figure 10, Figure 11, and Figure 12 test whether models can accumulate evidence, select the appropriate recipient, and return to silence after acting. In the cooking case, LIVEASSISTANT waits until repeated join requests form a clear host-facing need. In the education case, it stores an unresolved question and answers only after the instructor provides the explanation. In the outdoor case, it delays a motion-related answer until sufficient visual evidence distinguishes walking from jogging. General baselines often answer prematurely, continue responding after the need has passed, or remain silent throughout the trajectory. These behaviors are consistent with their lower state and recipient scores in Table 1.

Native streaming models. Figure 13, Figure 14, and Figure 15 focus on social and platform events. LIVEASSISTANT detects persistent gift activity, retains transient events, and distinguishes viewer requests from host-facing notifications. Native streaming baselines primarily remain silent or describe the visible scene, and some produce repetitive or unrelated text. They rarely convert

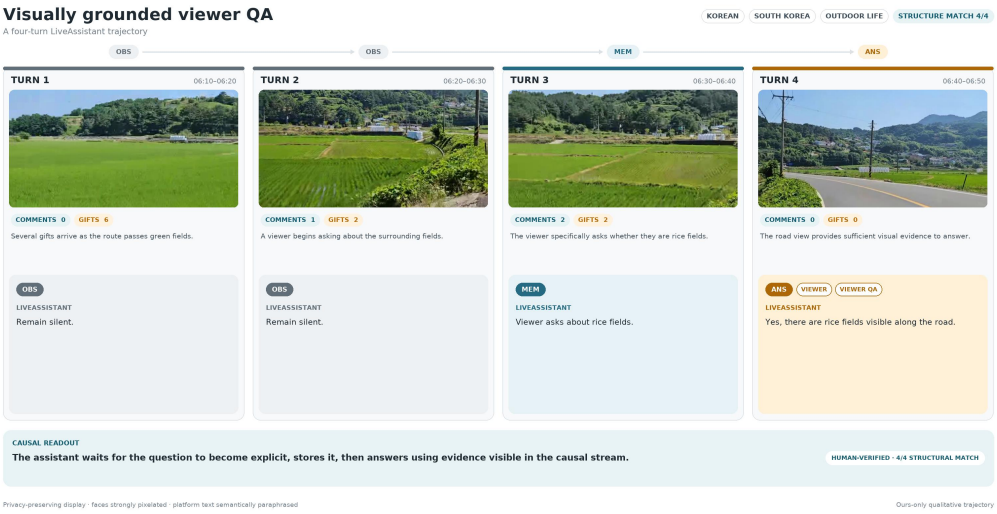


Figure 8: Korean outdoor stream. LIVEASSISTANT retains a viewer question until the rice fields become visually identifiable.

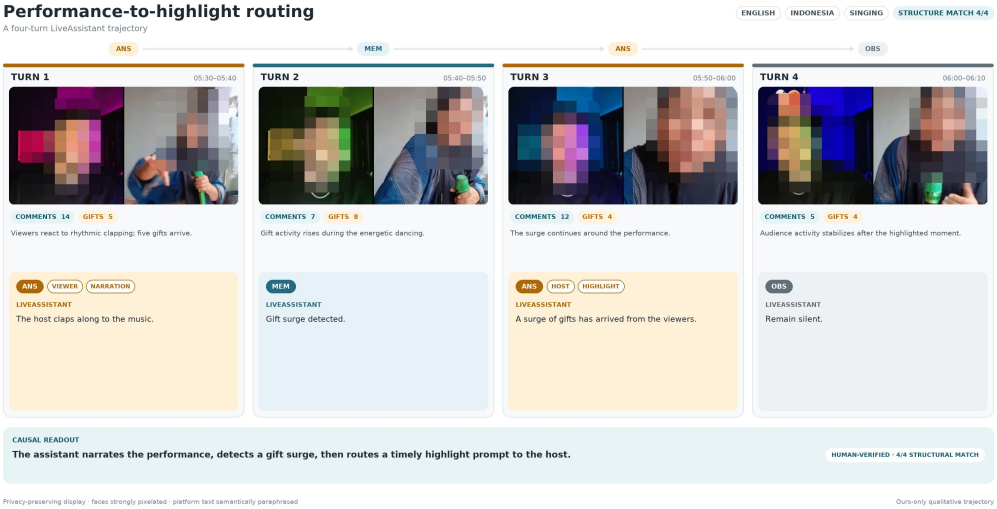


Figure 9: English singing. LIVEASSISTANT transitions from viewer narration to a host-facing gift-surge notification.

comments and gifts into structured recipient-specific actions, matching the gap in task and routing performance reported in Table 1.

Summary. Across all six cases, the main difference lies in decision structure rather than surface fluency. General models frequently treat each chunk as an independent opportunity to answer, while native streaming models focus on passive perception. LIVEASSISTANT coordinates OBS, MEM, and ANS across turns and grounds each external action in an evolving social need.

D.3 QUALITATIVE FAILURE ANALYSIS

We examine four four-turn failures involving semantic grounding, repeated comments, source attribution, and action timing. Each case preserves consecutive 10-second inputs, platform signals, and the model's preceding outputs.

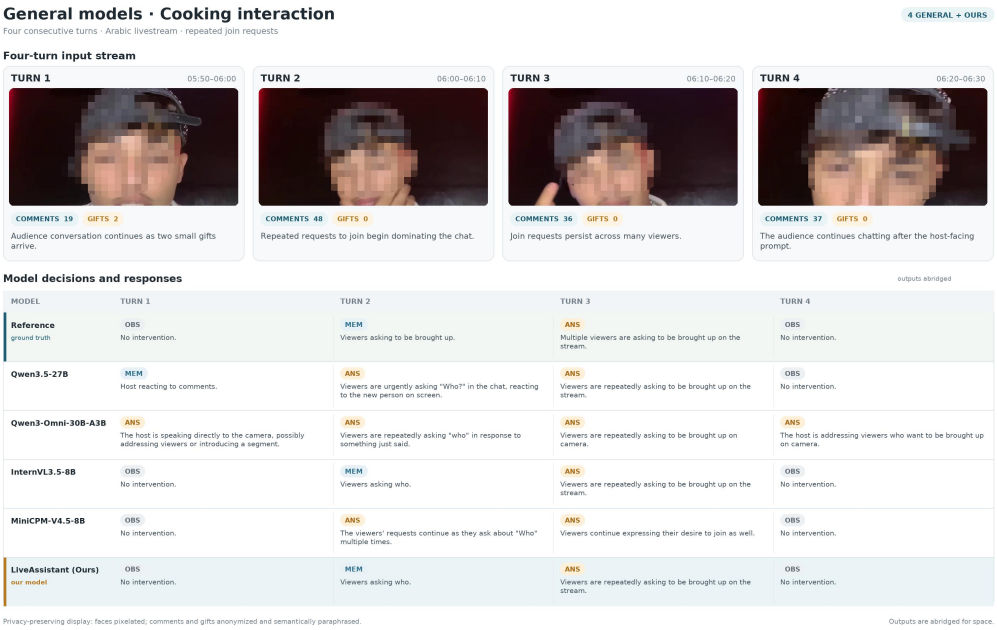


Figure 10: Cooking case. LIVEASSISTANT waits for repeated requests before notifying the host.

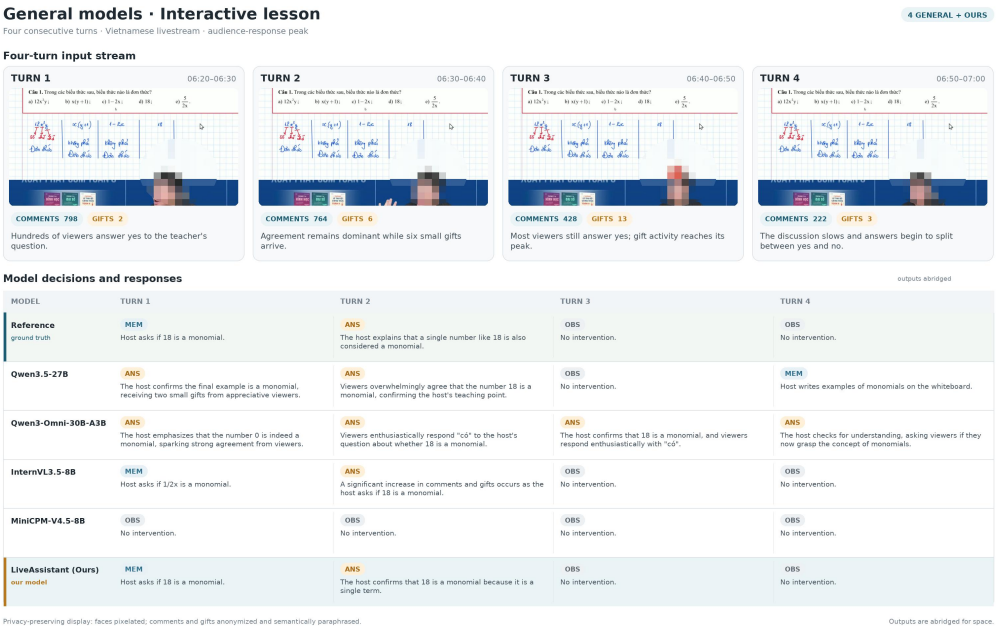


Figure 11: Education case. LIVEASSISTANT answers after the instructor provides sufficient evidence.

Semantic reliability. Figure 16 shows an Arabic cooking stream in which LIVEASSISTANT matches the reference state in all four turns but misinterprets the content. It maps a discussion about contacting TikTok to a returning guest and repeated join requests to an unrelated statement about the host, showing that correct action structure does not guarantee grounded content. In Figure 18, comments claim that the host received a hat gift while the platform counter remains zero.

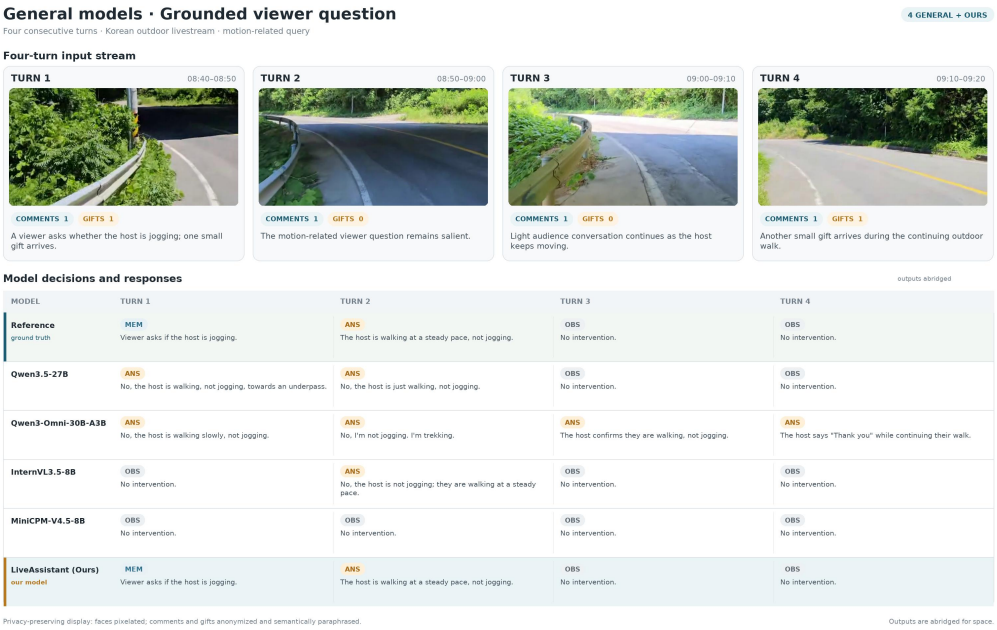
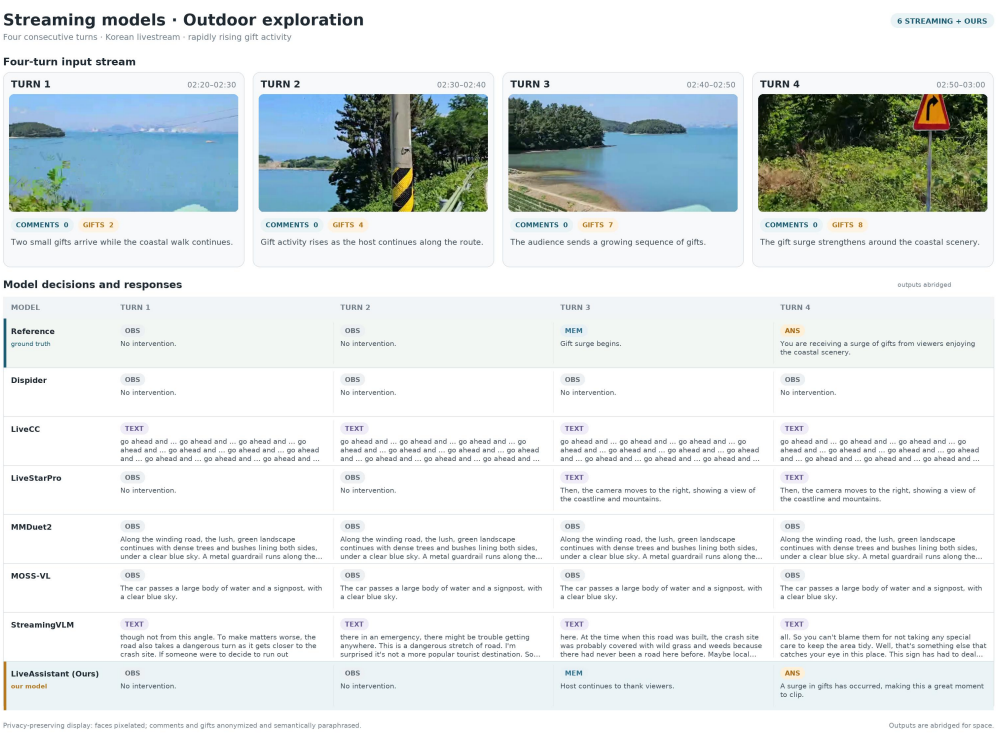


Figure 12: Outdoor question case. LIVEASSISTANT retains the query and waits for visual confirmation.

LIVEASSISTANT treats the comment as a verified gift event and notifies the host, collapsing an unverified viewer claim into platform state.

Decision calibration. In Figure 17, repeated answers in a Vietnamese math stream push the policy from silence to response after the question has already been resolved. The model follows comment frequency rather than information novelty and emits a redundant answer. In Figure 19, LIVEASSISTANT escalates a repeated MVP question while the answer remains unavailable. The question should remain in memory until new evidence or an appropriate interruption point emerges.

Summary. These failures separate semantic reliability from decision calibration. LIVEASSISTANT can select the correct action state while grounding the content incorrectly, and it can recognize a valid social signal while responding at the wrong time. Stronger multilingual grounding, source-aware reasoning, novelty detection, and interruption control remain the main directions for improvement.


Streaming models · Outdoor exploration
6 STREAMING + OURS
Four consecutive turns · Korean livestream · rapidly rising gift activity
Four-turn input stream
TURN 1 02:20–02:30
COMMENTS 0 GIFTS 2
Two small gifts arrive while the coastal walk continues.
TURN 2 02:30–02:40
COMMENTS 0 GIFTS 4
Gift activity rises as the host continues along the route.
TURN 3 02:40–02:50
COMMENTS 0 GIFTS 7
The audience sends a growing sequence of gifts.
TURN 4 02:50–03:00
COMMENTS 0 GIFTS 8
The gift surge strengthens around the coastal scenery.
Model decisions and responses
outputs abridged
MODEL
TURN 1
TURN 2
TURN 3
TURN 4
Reference
ground truth
OBS
No intervention.
OBS
No intervention.
MEM
Gift surge begins.
ANS
You are receiving a surge of gifts from viewers enjoying the coastal scenery.
Dispider
OBS
No intervention.
OBS
No intervention.
OBS
No intervention.
OBS
No intervention.
LiveCC
TEXT
go ahead and ... go ahead and ... go ahead and ...

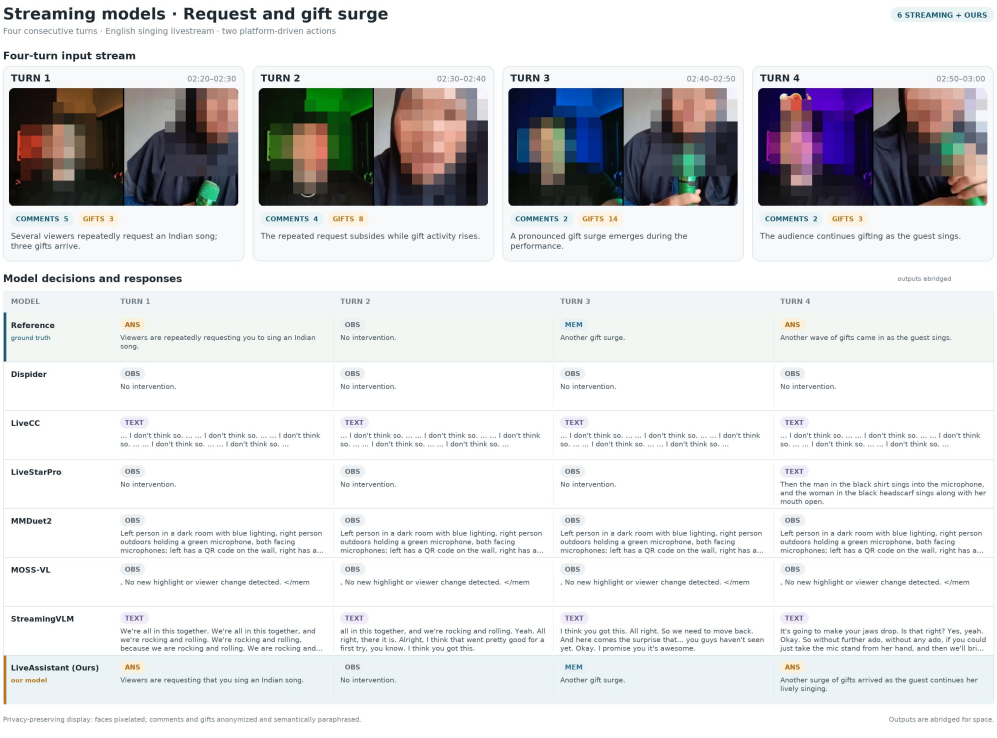


Figure 15: Request and gift case. LIVEASSISTANT handles a viewer request and a later gift surge.

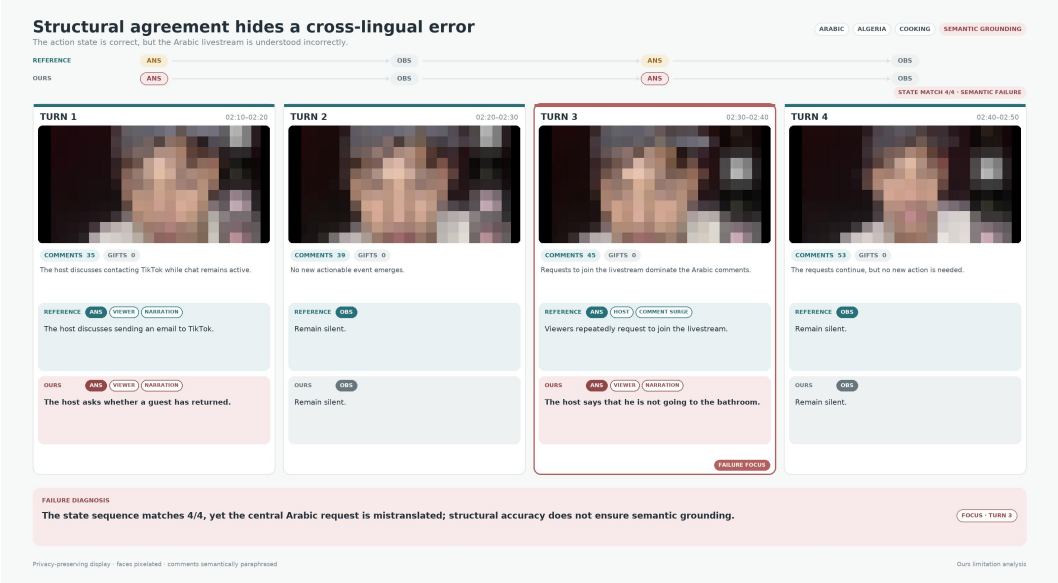


Figure 16: Cross-lingual grounding failure. The state sequence matches all four turns, but LIVEASSISTANT misinterprets the Arabic speech and comments.

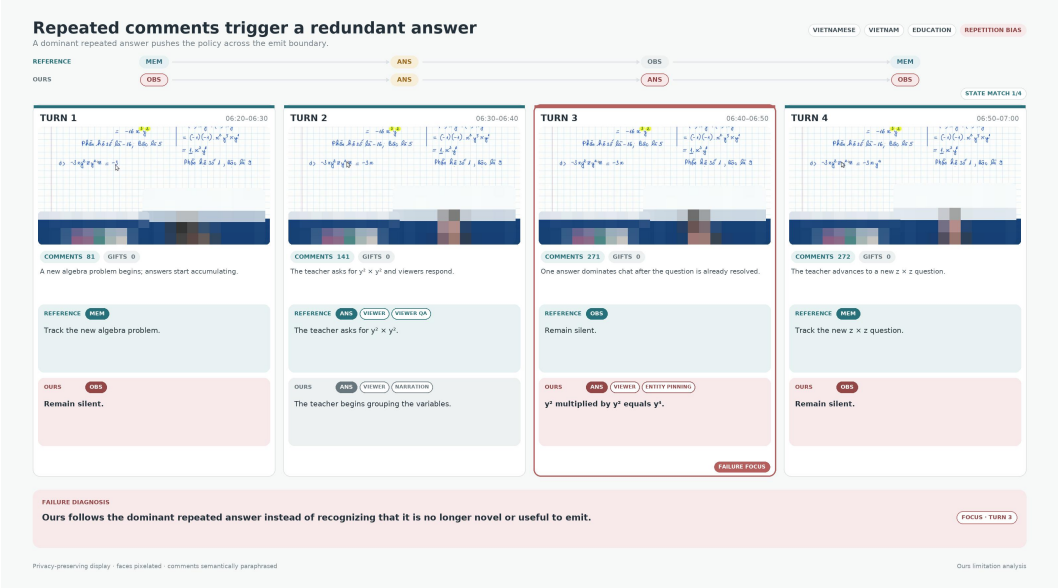


Figure 17: Comment-repetition failure. Repeated comments trigger a redundant answer after the question has been resolved.

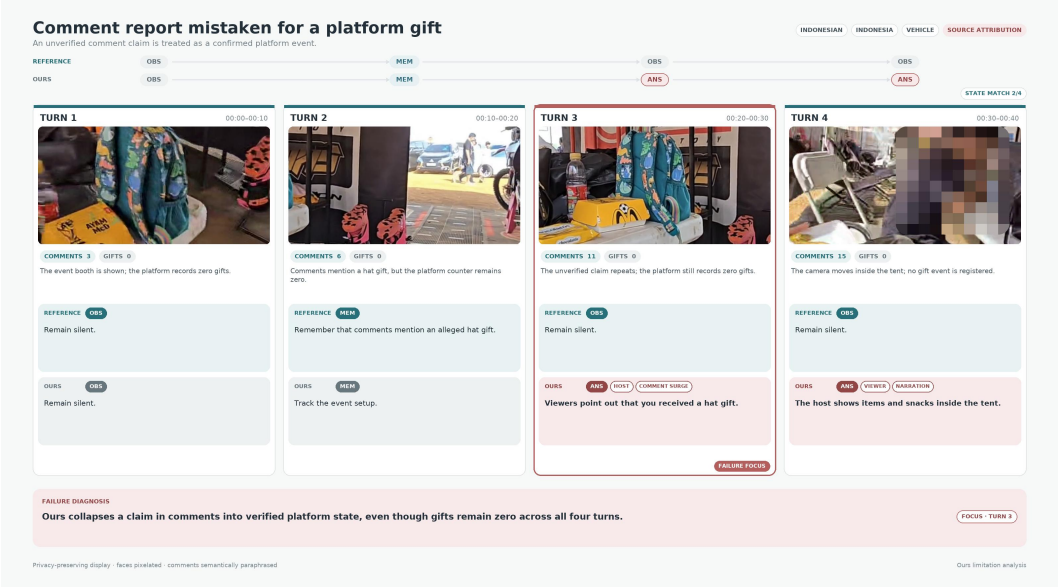


Figure 18: Gift-attribution failure. LIVEASSISTANT treats an unverified comment as a confirmed gift despite a zero platform count.

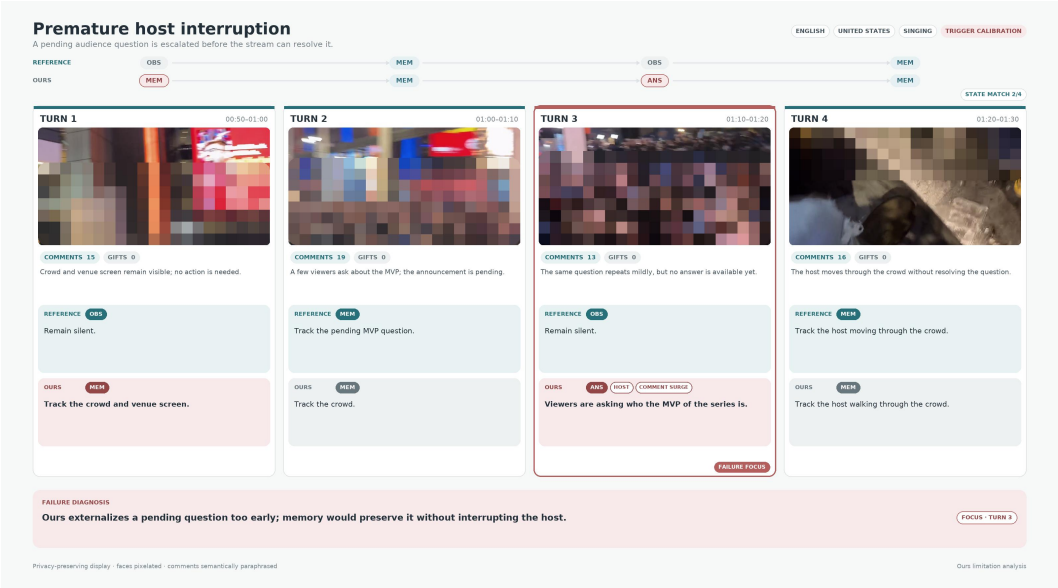


Figure 19: Host-interruption failure. LIVEASSISTANT externalizes a pending audience question before an answer or suitable interruption point emerges.