STREAM-OF-THOUGHT: ALWAYS-ON REASONING FOR REACTIVE AND PROACTIVE ASSISTANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Live multimodal assistants, the systems that continuously stream a user's egocentric viewpoint, make persistent perception of the user's environment practical. Yet their interaction paradigm remains *reactive*: reasoning begins only once a user speaks, and the incoming stream goes unexamined until then. We propose **Stream-of-Thought** (SoT), which enables an online vision-language model to simultaneously and adaptively emit thought tokens as it perceives a streaming input, and to decide on its own when to turn those thoughts into speech. Thoughts cached across a session serve both modes of assistance: (1) *reactive*, where complex questions are answered faster and long-horizon tasks are tracked because much of the required reasoning is already done, and (2) *proactive*, where the same cached thoughts ground the help that the user never asked for. Since unconstrained "daydreaming" only lengthens context, we train thought to be *utility-directed*, optimal under both objectives: a two-stage SFT over a curated SoT-SFT dataset, followed by **Think–Speak Group Policy Optimization** (TSGPO), a novel RL algorithm that credits thinking and speaking separately. We further introduce **UnpromptedBench**, a manual collection of unprompted, open-ended assistance in egocentric streams that tests what is said and when. Through extensive evaluations, we show that a single SoT model combines strong reactive *and* proactive capabilities under a tight latency budget, outperforming prior streaming systems at comparable scale and rivaling far larger single-regime specialists.

1 INTRODUCTION

The advent of smart glasses has made it possible for an assistant to perceive a user's visual environment continuously (Engel et al., 2023; Grauman et al., 2022). Yet the prevailing interaction paradigm remains *reactive*: reasoning begins only once a user initiates a query (Bai et al., 2025b;a; Feng et al., 2026), and the streamed input is treated as passive, accumulated context to be attended over on demand. Our vision instead centers on an *omni-observant* assistant, a system that continuously ingests visual streams and conversation (Chen et al., 2024a; Lu et al., 2026), and thinks in the background as they unfold. This persistent awareness lets the assistant build up an understanding of what the user is doing, anticipate what they will need, and offer help at the moment it is useful.

To this end, we propose **Stream-of-Thought** (SoT), an approach that enables an online VLM (Qian et al., 2025; Xu et al., 2026) to continuously analyze a changing environment by simultaneously emitting thought tokens, and to independently decide when to turn those thoughts into action (Wang et al., 2024; 2026d). Formulating *thinking and watching* on a duplex clock in this way yields two practical benefits. First, thoughts cached over the course of a session allow the model to answer complex questions far faster and to track long-horizon tasks (Guan et al., 2026; Liu et al., 2026b; Yang et al., 2026b), since much of the necessary reasoning has already been done. Second, the same cached thoughts serve as grounding context for *proactively* offering help, even when the user has issued no written or spoken query.

The success of this approach hinges on guiding the model to think in a *productive*, *utility-directed* manner: predicting plausible questions the user might ask later, and constantly seeking ways to help before being asked. Unconstrained "daydreaming," by contrast, only lengthens the context without improving what the model can do with it (Chen et al., 2024b; Sui et al., 2025). We therefore shape the thought stream in two stages. We first formulate an SFT stage over our curated SOT-SFT dataset, through which the model learns to think continuously along directions that retrospectively

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.

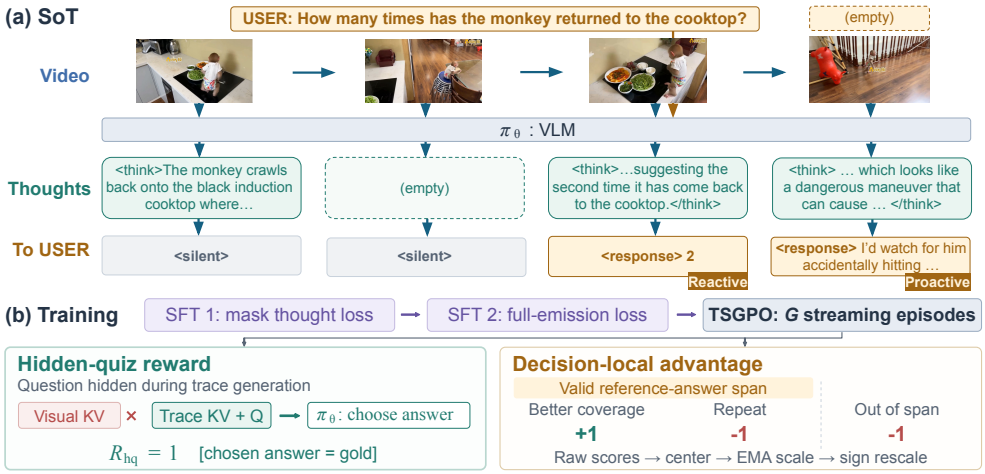


Figure 1: (a) **SoT overview**. The model adaptively decides when to think, and speaks to user when a query is given (*reactive*), or when evidence is ready for assistant-initiated interruptions (*proactive*). (b) **Training pipeline** involves a 2-stage SFT (§2.1) followed by our proposed TSGPO RL (§2.2).

prove helpful in answering questions. We then devise an RL strategy over complete streaming episodes and propose **Think-Speak Group Policy Optimization** (TSGPO), a novel RL algorithm that assigns credit separately to the two decisions a streaming model makes. TSGPO introduces (i) a *decision-local advantage*, which attributes every reply's marginal contribution to the emission token that produced it rather than diffusing a single episode-level return across the trajectory (Shao et al., 2024), and (ii) a *deferred-query objective*, which scores a reasoning trace by the answerability of a question withheld throughout its generation.

We evaluate our model on both (a) **reactive** streaming video tasks (*i.e.* user initiates a complex query at the end of a video segment) (Niu et al., 2025; Lin et al., 2026) and (b) **proactive** scenarios (*i.e.* assistant interrupts at the right moment during the video) (Wang et al., 2025; 2026d). The biggest bottleneck in measuring proactive capability is the lack of an open-ended benchmark with reliable ground-truth. Previous literature is largely confined to procedural task datasets (Bao et al., 2023; Panchal et al., 2024; Bhattacharyya et al., 2025; Kundu et al., 2026), where the goal is fixed in advance and the right moment to intervene follows from a script, or to synthetically curated annotations (Wang et al., 2024; 2025; Zhang et al., 2025b), whose intervention points inherit the biases of the model that generated them. Therefore, we introduce **UnpromptedBench**, a benchmark of unprompted, open-ended assistance in egocentric streams in which every moment calling for intervention is directed by human annotators. **Our contributions** are as follows:

- We introduce **Stream-of-Thought**, a streaming model that generates reasoning traces while perceiving online video input, and decides on its own when to speak. We train it with SFT on a curated SoT-SFT dataset followed by **TSGPO**, an RL algorithm tailored to the think-speak decision structure of streaming episodes.
- We introduce **UnpromptedBench**, a human-annotated benchmark of unprompted, open-ended assistance in egocentric streams, which evaluates what an assistant says, when it says it, and whether that timing is justified by what the stream has revealed so far.
- Extensive experiments show that our approach enables a single model to continuously generate thoughts to aid reactive and proactive capabilities, which prior work addresses only in isolation.

2 Stream-of-Thought (SoT)

Problem Formulation. We consider a streaming video interaction where a video arrives as a sequence of segments $\mathcal{V} = (v_1, v_2, \ldots, v_T)$, each spanning s seconds. At step t the model observes segment v_t together with an optional user query q_t, with $q_t = \emptyset$ when no query is present. At every step the model emits a single token sequence with a fixed layout, $y_t =$ `<think>` z_t `</think>` g_t u_t:

- The reasoning step z_t may be empty. It is never shown to the user and never scored as output, but it remains in the context at every later step.
- The gate token $g_t \in \{\texttt{<silent>}, \texttt{<response>}\}$ declares whether the model speaks at step t.
- The utterance u_t is user-facing text, with $|u_t| = 0$ whenever $g_t = \texttt{<silent>}$.

Write $\mathcal{C}_t = (v_{1:t},\, q_{1:t},\, y_{1:t-1})$ for the context available at step t: every segment and query up to the current step, together with every emission before it. The model is an autoregressive vision–language model π_θ with form $\pi_\theta(y_t \mid \mathcal{C}_t) = \prod_{n=1}^{|y_t|} p_\theta\big(y_t^{(n)} \mid \mathcal{C}_t,\, y_t^{(<n)}\big)$. An episode is the concatenation $\mathbf{y} = (y_1, \ldots, y_T)$. Every part of y_t is produced by the same language-modeling head under ordinary next-token prediction over the existing vocabulary.

There are two decisions that prior streaming models conflate. The first factor decides whether to reason about the current segment, and the second decides whether to speak. Both are discrete and observable in a rollout, admitting individually attributable credit (§2.2).

Training Strategy. We train π_θ in two stages, with SFT alleviating cold start, and RL optimizing properties that per-step maximum likelihood cannot express, *e.g.*, whether a given reply was worth making, and whether a reasoning step recorded information that a later query will need.

2.1 Staged SFT

Learning the adaptive reasoning and speaking behaviors together proves difficult. We found a single SFT stage over the full emission is weak and provides a poor starting point for RL (see §5.2). We therefore separate the two into a two-stage schedule whose stages differ only in which tokens carry loss, i.e., $\mathcal{L}_{\mathrm{SFT}} = -1/|\mathcal{M}| \sum_{n \in \mathcal{M}} \log p_\theta\big(\mathbf{y}_n \mid \mathbf{y}_{<n}\big)$. The first stage excludes from $\mathcal{M}$ every position strictly inside a reasoning span, fixing when a reasoning step opens and closes and which gate follows it without committing the model to any particular reasoning style. The second sets $\mathcal{M}$ to every position, fitting the content of a trace once its placement is settled.

Prior work (Shi et al., 2025) schedules the same tension in the opposite direction, training on the full sequence first and then on the answer alone. We show that our order is stronger because an SoT episode interleaves a reasoning step and a gate decision at every step, leaving far more format to settle before content can be fitted to it. It also determines what RL inherits: a schedule whose final stage leaves the trace outside the loss ends with a reasoning decision the policy no longer varies, and a group-relative objective has nothing to compare at a decision that never varies (§2.2). The stage that closes the channel cannot be the last one.

2.2 Think–Speak Group Policy Optimization (TSGPO)

We optimize with GRPO (Shao et al., 2024), which samples G episodes $\{\mathbf{y}^{(i)}\}_{i=1}^{G}$ per prompt, scores each with a scalar return $R^{(i)}$, and standardizes within the group to $\hat{A}^{(i)} = (R^{(i)} - \mu_R)/\sigma_R$, applying that one number to every token of episode i.

A few challenges are specific to our scenario. First, GRPO gives every token of an episode the same advantage, so the gradient adjusts how often the policy speaks far more readily than where it speaks, which is particularly an issue for proactive responses, i.e., the model can raise its chance of having spoken at the right moment simply by emitting more. Second, when handling reactive problems, the policy must produce its reasoning traces before the user query arrives, with no way of knowing what it will be asked. Additionally, at each step the model must decide not only what to think about, but whether to think at all. TSGPO addresses these challenges separately.

Reward design. We first introduce a correctness reward R_{task}. In the reactive case, the episode carries one query and R_{task} is exact match against the verifiable answer or an LLM judgement for open-ended QA. In the proactive case the episode carries several reference turns, each with a span of segments in which it may be answered, and R_{task} averages over turns of how much of each reference the replies inside its span recover between them. Either way the signal is deliberately coarse, indicating whether the episode ended in the right place. However, optimizing z_t is difficult in the reactive case, where z_t is written before the query it may have to serve, and with R_{task} alone the policy leans on the visual context instead. While z_t are initially poor, the visual route is a cheaper way to raise R_{task}, so the reward is satisfied without z_t ever improving.

We therefore score a proxy we can observe: whether a trace will be effective later is not measurable at write time, but whether it answers a question given to the model afterwards *is* measurable. Let k be the step at which the episode's query arrives. We score the episode there with every visual token removed from attention, leaving the reasoning steps $z_{1:k-1}$ and the query as the only evidence in context. We restrict the reward to verifiable answers, hence the next token generated is the answer itself. One forward pass gives the distribution at that position, which we restrict to the item's option set O and renormalize, making a malformed answer impossible:

$$R_{\mathrm{hq}} = \mathbf{1}\Big[\ \arg\max_{o \in O}\ p_\theta\big(o \mid z_{1:k-1},\, q\big) = o^\star\ \Big] \tag{1}$$

where $o^\star$ is the gold option. It is informative exactly where R_{task} is not: withholding the video closes off the cheaper route and leaves the trace as the only lever on the score. Appendix A compares this hard decision against the soft alternative $p_\theta(o^\star \mid \cdot)$. This is design is much like a student preparing for a quiz without knowing what will be asked, which is why we call it the *hidden-quiz* reward. The final reward is a weighted sum, with ω weighting the hidden-quiz term:

$$R = R_{\mathrm{task}} + \omega\, R_{\mathrm{hq}} \tag{2}$$

Decision-local advantage (DLA). Under a single scalar reward covering the whole episode, speaking more is a cheap way to score better, as noted above for proactive cases. We observe that RL inflates the speaking rate by more than an order of magnitude over the supervised model it starts from. One may argue to add penalties on the reply count, but we found such terms hurt the model's potential capability on speaking contents (§5.2). What is missing is not a counterweight but an attribution, and we seek more finegrained credit assignment to achieve this. Specifically, we score each reply by what it adds, rather than by whether it is right. Let $\mathcal{S}_j \subseteq \mathcal{G}_i$ collect the replies falling inside turn j's span, and u_n how much of that reference the reply at gate n recovers. Then

$$c_n^{(i)} = \begin{cases} +1, & n \in \mathcal{S}_j \ \text{ and } \ u_n > \max_{m \in \mathcal{S}_j,\, m<n} u_m, \\ -1, & \text{otherwise}, \end{cases} \tag{3}$$

taking $\max \emptyset = 0$. A reply is credited only when it recovers more of the reference than anything said earlier in the same span, so restating the delivered content gets scored -1 however accurate it is, as does speaking inside no span at all. We add this score to the advantage at the gate token that produced the reply, i.e., $A_n^{(i)} = \hat{A}^{(i)} + \kappa\, \bar{c}_n^{(i)}$ and $\bar{c}_n^{(i)} = 0$ for $n \notin \mathcal{G}_i$, where $\mathcal{G}_i$ collects the gate positions of episode i that carry `<response>`, κ is a constant. $c_n^{(i)}$ is computed via three steps. Each rollout is first centered on its own mean, so that credit is redistributed within a rollout rather than added to it:

$$\tilde{c}_n^{(i)} = c_n^{(i)} - \frac{1}{|\mathcal{G}_i|} \sum_{m \in \mathcal{G}_i} c_m^{(i)}, \qquad \sum_{n \in \mathcal{G}_i} \tilde{c}_n^{(i)} = 0. \tag{4}$$

Credit is thereby separated from how much the policy says, *i.e.*, the centering leaves a rollout's total credit at zero, so κ redistributes credit among its replies rather than adding to them, and speaking more cannot raise the reward by itself. Equation 6 below breaks this balance deliberately, by a controlled amount. The centered score is divided by the standard deviation tracked across batches as a moving average:

$$\sigma_b = \rho\, \sigma_{b-1} + (1-\rho)\, \tilde{\sigma}_b, \qquad \hat{c}_n^{(i)} = \tilde{c}_n^{(i)} \big/ \max(\sigma_b,\, \epsilon), \tag{5}$$

with smooth coefficient ρ, floor ϵ, and $\tilde{\sigma}_b$ measured over the centered scores of the whole group at batch b. This makes κ dimensionless and comparable across episodes.

However, adding and removing probability mass to a gate token are not mirror operations, thus we propose to rescale the negative score:

$$\bar{c}_n^{(i)} = \begin{cases} \hat{c}_n^{(i)} \cdot \min\big(\alpha\, \pi_n/(1-\pi_n),\ \tau\big), & \hat{c}_n^{(i)} < 0, \\ \hat{c}_n^{(i)}, & \text{otherwise}, \end{cases} \tag{6}$$

where π_n is the policy's probability of the token it emitted at that gate, and τ caps the ratio.

Together, these settle the policy at a speaking rate rather than pushing it steadily in one direction. Positive and negative credit come into balance at an emission probability $\pi^\star = 1/(1+\alpha)$, and because κ multiplies both, it drops out of that balance. The credit is not the only gradient at the gate. However, the GRPO advantage and the KL penalty also act there, in opposite directions, and their influence relative to the credit falls as κ rises. Therefore, α names the rate and κ decides how closely it is met. More details are covered in Appendix B.

Reasoning-rate collapse. At every step the model chooses whether to write a reasoning trace or leave it empty. We found that the rate at which the model reasons drifts to one extreme or the other, and neither extreme is recoverable, since a decision the model no longer varies leaves a group-relative estimator nothing to compare. The choice is a single token, the one right after `<think>`, which either closes the span at once or starts a trace. The reasoning rate averages the probability of not closing over $\mathcal{D}$, which collects one such position per step, with logits ℓ_t:

$$p_t = 1 - \mathrm{softmax}(\ell_t)\big[\texttt{</think>}\big], \qquad \hat{r} = \frac{1}{|\mathcal{D}|}\sum_{t\in\mathcal{D}} p_t. \tag{7}$$

We keep the decision away from either extreme by maximizing its entropy,

$$\mathcal{L}_{\mathrm{reg}} = -\beta_H \cdot \frac{1}{|\mathcal{D}|}\sum_{t\in\mathcal{D}} H\big(p_t\big), \tag{8}$$

with coefficient β_H. An entropy bonus has proven to be a standard exploration device (Williams & Peng, 1991). Here it applies to a binary decision at a position the layout already fixes, and $H(p_t) \le \log 2$ bounds it by construction. The bonus also maximizes the probability that a group samples both choices at the decision, which is what makes a group-relative advantage informative there (Appendix C). A variant holds $\hat{r}$ inside a fixed band under a two-sided quadratic hinge (§5.2).

Full objective. Combining the three components, TSGPO minimizes:

$$\mathcal{L}_{\mathrm{TSGPO}}(\theta) = -\sum_{i=1}^{G}\sum_{n} A_n^{(i)} \log \pi_\theta\big(\mathbf{y}_n^{(i)} \mid \mathbf{y}_{<n}^{(i)}\big) \;+\; \beta\, D_{\mathrm{KL}}\big(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big) \;+\; \mathcal{L}_{\mathrm{reg}}, \tag{9}$$

Neither channel requires an additional rollout. Training replays each episode in a single differentiable pass under the same visual window used at inference, thus the loss on an answer propagates into the representations of the reasoning steps that preceded it.

3 Data Curation

SFT. We leverage Qwen3-VL-235B-A22B-Thinking (Bai et al., 2025a) to write traces for each video segment, drawing each trace as $z_t \sim \mathcal{A}(\cdot \mid v_{1:t},\, q_{1:t},\, z_{1:t-1})$. No term depends on a later segment, preventing the leakage of future information, and $q_{1:t}$ is empty at every step before the question is asked. The trace is therefore question-blind by construction rather than by instruction, and holds exactly what π_θ holds at the same point in the episode.

We curate the trace for each video segment to link something seen in the past, optionally with what it implies given the record so far. The implication is on the substance of the trace, *i.e.*, it asserts something that appears in no frame and exists only across them. The curation model reaches it by looking back over the trace for one of five relations, *i.e.*, a recurrence, a completion or abandonment, a reversal, a change since the last mention, and something absent for long enough that its absence in and of itself is informative. An entry may also be a bare occurrence if no implication is available. Not every segment earns a thinking token. What earns one is an occurrence, *e.g.*, something started, finished, appeared, or changed. To ensure our generated traces are useful for answering user's question, we use Kimi-K2.6 (Moonshot AI, 2026) as the judge to determine if the answer can be derived from the these traces, and iterate different traces until the answer is derived.

RL. We mix reactive and proactive data during RL, where reactive samples carry one question with a verifiable answer, while proactive samples have episodes carrying several reference answers given an initial query. We additionally include a small set of negative samples, asking something that never happens, to penalize the model for hallucinating for any plausible answers. For proactive data, it is fundamentally difficult to expect the model to deliver the answer at exact timestamps. Therefore, we extend the timestamp with a small tolerance span, of which the length is measured by Kimi-K2.6. We include a more detailed curation process as well as the prompts used in Appendix D.

4 UnpromptedBench

Current streaming-video benchmarks are typically query-conditioned: they measure when enough evidence has arrived to answer a known request (Wang et al., 2025). An always-available assistant

instead makes a joint decision: whether to interrupt, when an intervention would help, and what new information or action to provide. UNPROMPTEDBENCH evaluates this setting in long first-person video, where correct silence is expected for most of a session.

Human ground truth. For session i, a system observes a causal stream $\mathcal{V}_i$ and may remain silent or emit interruption. Its reference is a set of human-approved opportunities $\mathcal{H}_i = \{(I_j, u_j, c_j)\}$, each specifying an acceptable time interval, intent, and category (procedural, safety, recall, recommendation, or social). Thus the target defines both *what* would help and *when*.

Candidate discovery combines a general whole-video pass with targeted safety and recall passes. The targeted prompts broaden coverage of rare but important intervention types, while the recall pass conditions proposals on an event log to reduce invented callbacks. The passes are merged and deduplicated; automatic filters provide hints but never define gold labels. The pool contained 611 candidates from 48 videos, each independently rated by three of 24 annotators (1,833 ratings). Ratings separately assess relevance/grounding, timing and future leakage, redundancy, and egocentric perspective (defined in Appendix G). Observed agreement was 60.7%, 65.2%, 88.5%, and 96.6%, respectively (Fleiss' $\kappa = .216, .222, .520, .145$). The video-disjoint split contains 454 development candidates in 36 videos and 157 held-out candidates in 12. Appendix G provides the full agreement, routing, exclusion, and provenance records. Figure 3 summarizes the complete workflow.

Scoring. We score each response in two steps. First, a judge asks whether the assistant should have spoken: the response must be grounded, correct, timely, useful, safe, and not a repeat of earlier speech. Every response remains in the denominator, so narration, unsupported claims, and repetition reduce precision. Second, for responses that pass, we ask whether they provide the same useful help as one of the human-identified opportunities available at that time. One response can cover at most one opportunity, and each opportunity can be counted once. With N_E emitted responses, N_G responses passing the first check, N_C covered opportunities, and $N_H = 237$ development opportunities, we report $P = N_G/N_E, C = N_C/N_H, F_{0.5} = 1.25PC/(0.25P + C)$.

The precision-heavy summary reflects the cost of unsolicited speech, while reporting P and C separately exposes conservative or overtalkative systems. A useful response outside the finite catalog can pass precision but cannot earn coverage; a silent system has zero coverage and $F_{0.5}$.

Evaluator verification and reporting. We froze judge prompts, decoding, sample IDs, and reporting criteria before one expert reviewed 100 displayed decisions: 50 response-quality decisions and 50 response–reference comparisons, stratified over their automatic labels. The expert saw the judge rationale and linked human annotations, so this is judge-visible verification, not a blinded or inter-rater study. The response-quality judge agreed on 42/50 cases (84.0%; $\kappa = .643$): the expert confirmed all 30 automatic rejections but rejected 8 of 20 automatic passes. For matching, agreement was 47/50 (94.0%; $\kappa = .860$). These diagnostics quantify errors in the automatic evaluator; the frozen, human-validated opportunity catalog remains the benchmark ground truth. Table 2 reports development results; Appendix H provides metrics, uncertainty, and provenance.

5 EXPERIMENTS

Setup. We build SoT from Qwen3-VL-Instruct-4B (Bai et al., 2025a). We first applied staged-SFT on 53K and 15K samples we curated for stage 1 and stage 2, respectively. Then we apply TSGPO on 7K curated data for RL. Following a similar strategy as (Guan et al., 2026; Liu et al., 2026b; Wu et al., 2026), we bound the visual context, i.e., a query token attends only to the visual tokens of the most recent segments, while all text tokens remain visible for the whole episode, to reduce the overall latency. We include more training configurations and details in Appendix §E.

We evaluate SoT on the representative reactive streaming video tasks – OVO-Bench (Niu et al., 2025), StreamingBench (Lin et al., 2026), as well as proactive tasks – ProactiveVideoQA (Wang et al., 2025), EgoProactive (Kundu et al., 2026), and our proposed UnpromptedBench (§4). For offline video understanding, we evaluate SoT on VideoMME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), and MVBench (Li et al., 2024).

Table 1: **Reactive streaming benchmarks** including OVO-Bench and StreamingBench. † indicates results we produced using released model weights – self-reported numbers otherwise.

			OVO-Bench				StreamingBench	
Model	Size	Evict	RT	BT	FT	Avg.	RTVU	Avg.
Qwen3-VL-4B	4B	No	67.33	54.51	37.12	52.99	72.24	62.12
MMDuet2	3B	No	64.08†	45.03†	48.34†	52.48	76.56†	64.52
Em-Garde	2B + 7B	Yes	63.00	52.20	42.22†	52.47	76.70	64.59
VST	7B	Yes	67.20	56.70	**54.00**	59.30	79.50	69.40
ThinkStream	3B	Yes	67.03	52.30	34.45†	51.26	75.00	63.13
StreamOPD	4B	Yes	**82.60**	56.00	41.06†	59.89	**83.90**	71.90
ATDM	7B	Yes	64.70	44.30	36.10	48.37	71.60	59.99
R3-Streaming	7B + 4B	Yes	71.89	51.27	50.60	57.92	76.36	67.14
TWW	8B	Yes	63.68	52.77	53.29	56.05	74.40	65.22
SoT	4B	Yes	72.16	**68.57**	52.40	**64.37**	81.27	**72.82**

Table 2: **Proactive benchmarks.** ProactiveVideoQA reports PAUC ↑ / repetition ↓; EgoProactive: GMean-F1 ↑. UnpromptedBench: precision/coverage/$F_{0.5}$ ↑ (all in %).

			ProactiveVideoQA		EgoProactive	UnpromptedBench
Model	Size	Evict	WEB	EGO	GMean-F1	P/C/$F_{0.5}$
Qwen3-VL-4B	4B	No	45.4/33.3	34.6/49.1	19.0	1.15/0.42/0.86
MMDuet2	3B	No	53.3/4.2	33.6/**8.1**	7.5†	0.01/0.00/0.00
Em-Garde	2B + 7B	Yes	44.3/4.5	**52.3**/17.4	50.7†	9.40/0.42/1.79
VST	7B	Yes	42.5/37.9†	36.7/64.1†	11.8†	7.79/0.00/0.00
ThinkStream	3B	Yes	26.9 / 9.8†	20.5 / 19.1†	35.9†	0.00/0.00/0.00
StreamOPD	4B	Yes	40.5 / 10.6†	36.1 / 20.7†	26.5†	28.57/0.00/0.00
SoT	4B	Yes	**54.3 / 4.1**	46.5/8.7	**57.6**	**45.81/2.11/8.91**

5.1 RESULTS

In this section, we compare our results not only against the baseline Qwen3-VL-4B, but prior works on streaming video understanding with thinking, including VST (Guan et al., 2026), ThinkStream (Liu et al., 2026b), StreamOPD (Wu et al., 2026), ATDM (Zhang et al., 2026b), R3-Streaming (Liu et al., 2026a), and TWW (Wang et al., 2026b).[1] We also include two recent works on proactive video understanding, namely MMDuet2 (Wang et al., 2026d) and Em-Garde (Zheng et al., 2026), for comparing our result on proactive benchmarks.

5.1.1 STREAMING VIDEO UNDERSTANDING

Reactive tasks (Table 1). We see that SoT significantly outperforms the Qwen3-VL-4B baseline across all tasks. Compared to prior works, SoT, as a single model, has demonstrated competitive results on every subtask and outperformed all these works when considering the overall score, including works with multiple models involved, with larger model size, and with stronger base model (i.e., StreamOPD starts with Qwen3.5 with thinking mode enabled). In particular, SoT has shown remarkably strong capability on backward tracking (BT) subtask, which confirms the effectiveness of both our proposed data curation pipeline and the hidden-quiz mechanism (see §5.2 for details).

Proactive tasks (Table 2). In ProactiveVideoQA, while PAUC and repetition are usually traded against each other, SoT shows a significant improvement on the balance of these two, showing competitive performance against the prior works that are specifically optimized for this benchmark or with a combination of multiple larger models. UnpromptedBench proves challenging for all models, which is expected given its strict metric using conservative semantic matching and redundancy-barred precision. Despite this, SoT exhibits significantly higher scores than all prior works.

[1]We were unable to find publicly released weights for ATDM, R3-Stream and TWW, so we could not produce their results on the proactive benchmarks, excluding them in Table 2.

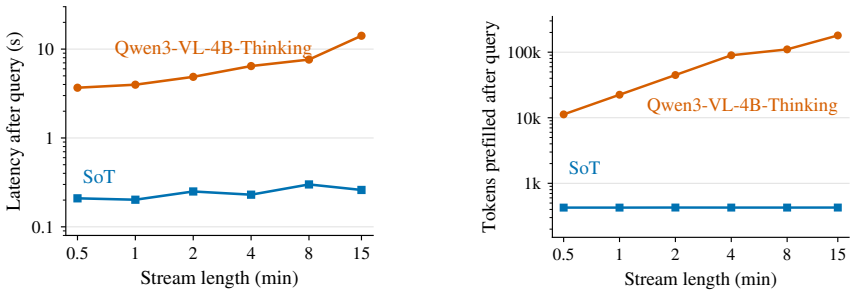


Figure 2: **Inference cost against stream length**. Latency after the query (left) and tokens prefilled after it (right), on logarithmic axes.

In summary, SoT equips a single model with both reactive and proactive capabilities, outperforming many existing works including larger single-regime specialists.

5.1.2 Inference efficiency

We measure what a user waits for, the time from the arrival of a query to the completion of its answer, and compare SoT with the reasoning counterpart of SoT's backbone, i.e., Qwen3-VL-4B-Thinking (Bai et al., 2025a), which runs offline as it was trained, receiving the video that precedes the query as a single context. We truncate eight OVO-Bench videos to six lengths between 30 seconds and 15 minutes, fixing the content and varying only the length of the stream (Liu et al., 2026b; Xu et al., 2026). Experiments are conducted on a single Nvidia H100.

Figure 2 summarizes the results. SoT answers more than $15\times$ faster than the thinking model at every length, with a latency that stays relatively flat as the stream grows. This comes from what remains to be done once the query arrives. The thinking model must first prefill the entire video, up to $420\times$ more tokens than SoT, and only then begin to reason. SoT does both while the video plays, i.e., it encodes each segment as it arrives and adaptively reasons on them, thus all that remains is to prefill the newest segment and the query, and to write at most one short reasoning step. As a result, the wait is thus set by a single segment rather than by the length of the video.

5.1.3 Offline video understanding

We also examine how enabling the streaming capability affects the model's original capability on offline video understanding. We evaluated on Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), and MVBench (Li et al., 2024). Their results are summarized in Table 3. As seen, SoT almost preserves the offline performance to the baseline level – consistent with the observation in (Wang et al., 2026d).

Table 3: **Offline video understanding.** Reported against the base model on VideoMME, LongVideoBench, MVBench.

Model	VMME	LVB	MVB
Qwen3-VL-4B	69.3	60.4	67.9
SoT-4B	67.6	59.9	66.5

5.2 Ablation Study

Impact of Hidden-quiz Reward. Table 4 validates the effectiveness of the hidden-quiz reward on OVO-Bench. All compared arms have the data and every other reward term held fixed. We see that regardless of having this reward or not, both achieve reasonably good scores on RT and FT, but adding hidden-quiz demonstrates significantly better results for BT. For RT, related visual context is still in the KV cache, thus even if the reasoning is not perfect, the model can still extract answers from the visual context. For FT, the reasoning is question-conditioned, which is a much easier case. Comparatively, BT is more challenging as the quality of the traces directly determines the capability to derive the answer. As a result, we see hidden-quiz benefits more on BT, and with slight improvements on RT and FT.

Table 4: Effect of hidden-quiz reward.

	RT	BT	FT
SFT	71.21	60.73	46.85
w/o hidden-quiz	72.06	60.52	48.22
w. hidden-quiz	72.16	68.57	52.40

Table 5: Comparison of different SFT components, including three different scheduling strategies and an ablation on the effectiveness of our curated SFT data.

	SFT			RL
Schedule	Think %	OVO-Bench	StreamingBench	Think %
Qwen3-VL-4B w. SoT prompt	88.61	26.43	33.84	80.74
Standard SFT	59.01	56.41	78.12	57.20
SFTKey	0.00	59.21	79.10	0.00
Ours	59.93	59.60	79.72	56.32

SFT Components. Table 5 first compares three SFT schedules over the same corpus: a single stage with standard SFT, SFTKey (Shi et al., 2025), which first trains on all tokens, then the answer alone with reasoning traces masked out, and our proposed two-stage SFT with the opposite order. We then apply RL on top of them to observe the rate of non-empty thinking traces. An SFT checkpoint headed for RL has to be good in two ways at once, and Table 5 separates them. Numerically both SFTKey and our proposed staged SFT stand out, with ours achieving an even higher score. On the second criterion, whether the decisions the RL objective will act on are still being sampled, SFTKey's think rate collapses under our adaptive reasoning settings, which is problematic for RL given the nature of GRPO's group-relative estimator. Further analysis and derivation is in §C.

We then run our baseline Qwen3-VL-4B with our SoT prompt, which guides it to think as the video progresses. As seen, despite the high think rate, the model still sits at a poor OVO-Bench score. In effect, thinking is easy to ask for, but a trace that earns its place has to be learned. Leveraging our curated SFT data, a single-stage SFT substantially outperforms it.

6 Related Work and Conclusion

Streaming Video Understanding. Streaming video understanding poses two problems usually studied apart. The first is reactive: queries arrive mid-stream and must be answered under memory and compute that cannot grow with elapsed time. Prior work concerns what to retain through budgeted memories, eviction and cache management (Xu et al., 2026; Yang et al., 2026b; Chatterjee et al., 2025), and how to stay in real time (Ding et al., 2025). The second is proactive: the model decides when to speak, and systems differ in where that decision lives, *e.g.,* an auxiliary head over hidden states (Chen et al., 2024a; Qian et al., 2025; Wang et al., 2026a), a token from the LLM head (Zhang et al., 2025b; Lu et al., 2026; Kim et al., 2026; Kundu et al., 2026), a trigger read off decoding statistics (Yang et al., 2026a), or a directly optimized reward (Wang et al., 2026d; Li et al., 2026; Zheng et al., 2026; Liu et al., 2026a). The two are hardly solved together, and systems that reason while watching are evaluated reactively (Guan et al., 2026; Liu et al., 2026b; Wu et al., 2026). SoT emits both decisions from one model, with no auxiliary modules.

Video Reasoning. Reasoning in video have been exploited through RL with verifiable rewards (Feng et al., 2026; Li et al., 2025; Wang et al., 2026c; Maaz et al., 2025), much of it adapting the group-relative estimator with difficulty-aware advantage weighting (Park et al., 2026) or token-level informativeness weighting (Dang et al., 2026). All of it reasons once, after the question and with the whole video in context. A stream withdraws both conditions. Streaming work relaxes the first, overlapping watching with thinking (Zhang et al., 2026a; Wang et al., 2026b; Guan et al., 2026; Zhang et al., 2026b) or compressing traces into streaming memory (Liu et al., 2026b), but reasons at every step by construction, not by choice. Whether to reason at all has been studied only for single queries in text (Zhang et al., 2025a; Fang et al., 2026). A streaming episode poses that decision once per segment, and its trace must outlive the frames it describes and serve a question not yet asked. SoT treats reasoning as an action selected over time and optimized through our proposed TSGPO.

Conclusion. We introduce SoT, which reframes streaming assistance as *thinking while watching*, turning perception time into reasoning time to decide on its own when to speak. For reactive queries, the answer is largely reasoned out before the question is asked, masking its cost over streaming time rather than charging it at response time as chain-of-thought does. For proactive, the same trace gives the model a textual basis for its interruption decision, the modality VLMs are pretrained to reason over, rather than raw frames. Empirical results across reactive and proactive benchmarks confirm both effects, and show that a single model can hold both regimes at once.

AI USE STATEMENT

In this work, we used generative AI tools for text polishing, e.g., shortening, tightening, and improving the clarity, and code cleaning, e.g., refactoring, renaming and commenting code written by us. We have not used generative AI tools for research ideation, method design, and experimental design. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Everything needed to reproduce this work is reported in the paper, with the Appendix covering more details. Appendix E gives the full training configuration for both SFT and RL. Appendix D describes details of our data curation, including the prompt used. Appendix G and H describes the construction and annotation of our proposed benchmark.

Upon de-anonymisation we will release the model weights, datasets, and our codebase.

REFERENCES

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Yuwei Bao, Keunwoo Peter Yu, Yichi Zhang, Shane Storks, Itamar Bar-Yossef, Alexander De La Iglesia, Megan Su, Xiao Lin Zheng, and Joyce Chai. Can foundation models watch, talk and guide you step by step to make a cake? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Apratim Bhattacharyya, Bicheng Xu, Sanjay Haresh, Reza Pourreza, Litian Liu, Sunny Panchal, Pulkit Madan, Leonid Sigal, and Roland Memisevic. Can multi-modal LLMs provide live step-by-step task guidance? *arXiv preprint arXiv:2511.21998*, 2025.

Dibyadip Chatterjee, Edoardo Remelli, Yale Song, Bugra Tekin, Abhay Mittal, Bharat Bhatnagar, Necati Cihan Camgoz, Shreyas Hampali, Eric Sauser, Shugao Ma, et al. Streaming videollms for real-time procedural video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22586–22598, 2025.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18407–18418. IEEE, 2024a.

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

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, et al. Do NOT think that much for 2+3=? On the overthinking of o1-like LLMs. *arXiv preprint arXiv:2412.21187*, 2024b.

Jisheng Dang, Jingze Wu, Teng Wang, Xuanhui Lin, Nannan Zhu, Hongbo Chen, Wei-Shi Zheng, Meng Wang, and Tat-Seng Chua. Reinforcing video reasoning with focused thinking. In *European Conference on Computer Vision*, pp. 623–640. Springer, 2026.

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

Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, et al. Project Aria: A new tool for egocentric multi-modal AI research. *arXiv preprint arXiv:2308.13561*, 2023.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *Advances in neural information processing systems*, 38:151268–151295, 2026.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

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 *European Conference on Computer Vision*, pp. 80–99. Springer, 2026.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado Van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3796–3803, 2019.

Yifei Huang, Guo Chen, Jilan Xu, Mingfang Zhang, Lijin Yang, Baoqi Pei, Hongjie Zhang, Lu Dong, Yali Wang, Limin Wang, et al. Egoexolearn: A dataset for bridging asynchronous ego-and exo-centric view of procedural activities in real world. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22072–22086. IEEE, 2024.

Junho Kim, Hosu Lee, James M. Rehg, Minsu Kim, and Yong Man Ro. Stride: When to speak meets sequence denoising for streaming video understanding. In *European Conference on Computer Vision*, pp. 38–56. Springer, 2026.

Kaustav Kundu, Ritvik Shrivastava, Maxim Arap, Nanshu Wang, Xianhui Zhu, Quintin Fettes, Gautam Tiwari, Parth Suresh, Théo Moutakanni, Alejandro Castillejo Munoz, et al. Plan, watch, recover: A benchmark and architectures for proactive procedural assistance. *arXiv preprint arXiv:2606.04970*, 2026.

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.

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

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

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

Jinming Liu, Jianguo Huang, Zhaoyang Jia, Jiahao Li, Xiaoyi Zhang, Zongyu Guo, Bin Li, Wenjun Zeng, Yan Lu, and Xin Jin. An efficient streaming video understanding framework with agentic control. *arXiv preprint arXiv:2605.17921*, 2026a.

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. *arXiv preprint arXiv:2603.12938*, 2026b.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, et al. Aura: Always-on understanding and real-time assistance via video streams. *arXiv preprint arXiv:2604.04184*, 2026.

Muhammad Maaz, Hanoona Rasheed, Fahad Shahbaz Khan, and Salman Khan. Video-r2: Reinforcing consistent and grounded reasoning in multimodal language models. *arXiv preprint arXiv:2511.23478*, 2025.

Moonshot AI. Kimi k2.6, 2026. URL https://www.kimi.com/ai-models/kimi-k2-6.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913. IEEE, 2025.

Sunny Panchal, Apratim Bhattacharyya, Guillaume Berger, Antoine Mercier, Cornelius Bohm, Florian Dietrichkeit, Reza Pourreza, Xuanlin Li, et al. What to say and when to say it: Live fitness coaching as a testbed for situated interaction. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jinyoung Park, Jeehye Na, Jinyoung Kim, and Hyunwoo J Kim. Deepvideo-r1: Video reinforcement fine-tuning via difficulty-aware regressive grpo. *Advances in Neural Information Processing Systems*, 38:138605–138632, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24045–24055. IEEE, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Xiaofeng Shi, Qian Kou, Yuduo Li, and Hua Zhou. Rethinking supervised fine-tuning: Emphasizing key answer tokens for improved llm accuracy. *arXiv preprint arXiv:2512.21017*, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Hado P Van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. Learning values across many orders of magnitude. *Advances in neural information processing systems*, 29, 2016.

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. *Advances in Neural Information Processing Systems*, 38:132332–132359, 2026a.

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

Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capability in mllms via reinforced fine-tuning. *Advances in neural information processing systems*, 38:4350–4376, 2026c.

Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. Videollm knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format, 2024. URL https://arxiv.org/abs/2411.17991.

Yueqian Wang, Xiaojun Meng, Yifan Wang, Huishuai Zhang, and Dongyan Zhao. Proactivevideoqa: A comprehensive benchmark evaluating proactive interactions in video large language models. *arXiv preprint arXiv:2507.09313*, 2025.

Yueqian Wang, Songxiang Liu, Disong Wang, Nuo Xu, Guanglu Wan, Huishuai Zhang, and Dongyan Zhao. Mmduet2: Enhancing proactive interaction of video mllms with multi-turn reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 23257–23270, 2026d.

Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

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

Keming Wu, Baoyi Wang, Kaichen Zhang, Xiang An, Zuhao Yang, Sudong Wang, Haowei Zhu, Tingxuan Huang, Hongcheng Gao, and Bin Wang. Streamopd: A post-training recipe with spatio-temporal cue gating for streaming video understanding. *arXiv preprint arXiv:2608.16320*, 2026.

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

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. *Advances in Neural Information Processing Systems*, 38:31266–31304, 2026a.

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*, 2026b.

Liping Yuan, Jiawei Wang, Haomiao Sun, Yuchen Zhang, and Yuan Lin. Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding. *arXiv preprint arXiv:2501.07888*, 2025.

Zhiyuan Zeng, Jiameng Huang, Zhangyue Yin, Jiashuo Liu, Ziniu Li, Bingrui Li, Yuhao Wu, Yining Zheng, Ge Zhang, Wenhao Huang, et al. Balanced aggregation: Understanding and fixing aggregation bias in grpo. *arXiv preprint arXiv:2605.04077*, 2026.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, 2025a.

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

Kecheng Zhang, Zongxin Yang, Mingfei Han, Haihong Hao, Yunzhi Zhuge, Changlin Li, Zhihui Li, Xiaojun Chang, et al. Progressive online video understanding with evidence-aligned timing and transparent decisions. In *International Conference on Learning Representations*, volume 2026, pp. 5092–5121, 2026b.

Yichi Zhang, Xin Luna Dong, Zhaojiang Lin, Andrea Madotto, Anuj Kumar, Babak Damavandi, Joyce Chai, and Seungwhan Moon. Proactive assistant dialogue generation from streaming egocentric videos. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12055–12079, 2025b.

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

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. *arXiv preprint arXiv:2603.19054*, 2026.

A HARD VERSUS SOFT SCORING OF THE HIDDEN QUIZ

Equation 1 scores the answer the model would actually give, rather than the probability it assigns the gold option. The two differ in a way that matters here. Consider a four-way item whose gold answer is B, and a policy whose option distribution shifts over training:

	A	**B** (gold)	C	D	answer given	soft $p_\theta(o^\star)$	hard R_{hq}
before	0.40	**0.35**	0.15	0.10	A, wrong	0.35	0
after	0.45	**0.40**	0.10	0.05	A, wrong	0.40	0

The soft score rises while the option the model would choose is unchanged, so the episode is wrong before and after, and there is no benchmark that would register a difference. A policy can therefore increase the soft return by redistributing mass among options it already ranks in the same order, which is a change no evaluation observes. The indicator moves only when the trace changes which option leads, and is accordingly the reading we use.

B THE DECISION-LOCAL CREDIT PIPELINE

Why the credit scale is global where GRPO's is per group. The two normalizations serve different ends. The group standard deviation in $\hat{A}^{(i)}$ equalizes prompt difficulty, which is a property of the group and must be measured there; a degenerate group correctly contributes nothing, since rollouts that all scored alike express no preference. Equation 5 instead fixes a unit for κ, which should not move with sampling noise, and a batch whose credits happen to be degenerates carries no news about that unit. Per-episode centering sends a single-reply episode to exactly zero, so a batch in which every episode spoke once has zero credit spread while the channel's scale is unchanged. Dividing by that batch's own standard deviation would be undefined, and the moving average, floored at ϵ, carries the scale through. Dividing by a positive scalar also cannot change a zero sum, so Equation 5 preserves the invariant that Equation 4 establishes. The moving average follows PopArt (Van Hasselt et al., 2016), which (Hessel et al., 2019) extends so that one task's reward scale does not dominate a mixture. We omit its output-preserving correction, which exists to hold a value head's predictions fixed as the statistics move; TSGPO is critic-free and has no such head. Whether a mean is taken per sequence or per token is already known to matter for the loss, where the two aggregations induce different biases (Zeng et al., 2026), and Equation 4 asks the same question of the credit.

The sign asymmetry and its fixed point. Differentiating $\log \pi$ with respect to the emitted token's logit gives $(1 - \pi)$, the mass not yet on that token. For a useful reply, which should add mass, this is the right scale: there is less left to add as π approaches one. For a wasteful reply, which should remove mass, the same factor points the wrong way, shrinking to nothing exactly when the model already almost always emits and there is more to remove. The quantity that should govern removal

is the mass already present, π, and multiplying by $\pi/(1-\pi)$ converts the one into the other, with α scaling how strongly removal acts relative to addition.

The fixed point follows once the factor is multiplied by the gradient it modifies. A positive credit pushes upward with weight $(1-\pi)$; a negative credit carries the extra factor, and the two combine as $(1-\pi)\cdot\alpha\pi/(1-\pi)=\alpha\pi$, pushing downward with weight $\alpha\pi$. Equation 4 makes the positive and negative credit masses within an episode equal, so the net pressure vanishes where the two weights coincide, at $(1-\pi^\star)=\alpha\pi^\star$ and hence $\pi^\star=1/(1+\alpha)$. Since κ multiplies both weights it cancels from this condition, so within the credit channel it governs the speed of approach and not the destination. Equation 6 is therefore not sum-preserving, and that is the instrument rather than a side effect. Centring makes $\sum_n \hat{c}_n^{(i)}=0$, so writing $P=\sum_{\hat{c}_n>0}\hat{c}_n^{(i)}>0$ the negative credits sum to $-P$; scaling only the negatives by $s=\min(\alpha\pi/(1-\pi),\tau)$ leaves a total of $(1-s)P$, which is positive below the fixed point, negative above it, and exactly zero at $s=1$, that is at $\pi=1/(1+\alpha)$. The zero sum that Equation 4 establishes is thus restored precisely at the equilibrium, and away from it the residual is the restoring force toward it. This statement takes a common π across credited positions; with position-dependent π_n the same sign structure holds at each position, but the aggregate no longer factorises. The derivation assumes the cap in Equation 6 is inactive, which holds for $\pi\le\tau/(\tau+\alpha)$, that is $\pi\le 0.75$ at $\tau=6,\ \alpha=2$; the equilibrium at $1/3$ lies well inside this range, and the cap serves only to bound the penalty in the high-emission regime. It also carries a single π where the policy has a distinct π_n at each credited position. With heterogeneity the balance is not pointwise: the positively credited positions push upward with total weight $\sum(1-\pi_n)$ against $\alpha\sum\pi_n$ from the negatively credited ones, so equilibrium equates two set averages rather than fixing any position separately, and $\pi^\star=1/(1+\alpha)$ is the common value at which they coincide. This is also why π is read at credited gate positions in Table 6: the credit acts nowhere else, so those positions are exactly the population the balance is taken over.

Two further gradients reach the gate. The fixed point above accounts for the credit channel alone. Two further gradients reach the same gate token, and each moves it. The episode advantage $\hat{A}$ is applied to every token of the episode, the gate included, and although $\mathbb{E}[\hat{A}]=0$ the rollouts that score well are also the ones that speak more — we measure a correlation of 0.55 between answer correctness and reply count — so its contribution is proportional to $\mathrm{Cov}(\hat{A},\,n_{\mathrm{emit}})(1-\pi)$ and pushes the rate up. The reference penalty pushes down: differentiating the k_3 estimator at the emitted token gives $\partial\,\mathrm{KL}/\partial z=(1-\pi_{\mathrm{ref}}/\pi)(1-\pi)$, a restoring force toward the supervised rate. Writing b and k for the strengths of the two relative to the credit channel, total drift vanishes at

$$\pi^\star \;=\; \frac{1+b-k}{1+b-k+\alpha}, \tag{10}$$

which recovers $1/(1+\alpha)$ at $b=k=0$. The interior fixed point therefore survives the additional forces: it remains unique and stable, α still controls it monotonically, and because b and k enter with opposite signs they partially cancel.

What this implies for κ. Both b and k scale as $1/\kappa$, so κ does not act on the rate of approach alone: it also sets how closely the realized rate tracks $1/(1+\alpha)$. A large κ leaves the credit channel dominant and the closed form accurate, while a small κ lets the other two terms displace the operating point. Reading Equation 10 against measured rates puts $b-k$ at a few hundredths at the κ we use, and shrinking with α as fewer replies weaken the covariance that produces b. The simpler statement that κ governs only speed is thus the $\kappa\to\infty$ reading of Equation 10, and is the regime the reported model operates in rather than a property of the mechanism.

Why a price on emissions is not the instrument. The obvious alternative to attributing credit is to charge for it, adding $-\lambda\,n_{\mathrm{emit}}/T$ to the return. We tried five variants of this: an unbounded linear price, a group-relative form anchored on a percentile of the reply counts among the higher-scoring rollouts in the group and charging unsuccessful episodes at least as much as the longest successful one, two revisions of that with a floor, and a tie-breaker charging only the replies that add nothing. None of them holds a rate. The reason is visible in the form. The term is monotone in the reply count and blind to which reply earned the return, so at any λ it removes useful and redundant replies in whatever proportion they occur. Below a threshold the policy outgrows the price and the rate rises more slowly; above it the arithmetic inverts, since the penalty is unbounded while the task reward

Table 6: **Where the speaking decision settles.** The predicted fixed point $\pi^\star = 1/(1+\alpha)$ against the value actually reached, and the displacement $b-k$ that Equation 10 implies from it. π is read *at credited gate positions only*, so it is the policy's probability mass on speaking where it speaks, not the rate at which it speaks. Sweep run at $\kappa = 20$; PAUC (ω=0.5) ↑ / duplicate rate ↓, single judge.

	$\pi^\star$	Measured π	Implied $b-k$	WEB
$\alpha = 1$	0.500	0.52	+0.083	49.4 / 11.0
$\alpha = 3$	0.250	0.258	+0.043	42.5 / 6.8
$\alpha = 9$	0.100	0.096	−0.044	34.6 / 1.8

Table 7: **The two constants of the DLA.** κ scales the credit channel, α sets the asymmetry between charging and crediting a reply.

	WEB	EGO	OVO	RTVU
SFT	40.91 / 1.37	29.61 / 1.29	59.93	79.72
Varying κ at $\alpha = 2$				
$\kappa = 0$	56.9 / 17.8	47.2 / 22.1	64.31	81.02
$\kappa = 6$	54.3 / 4.1	46.5 / 8.7	64.37	81.27
$\kappa = 12$	52.1 / 3.8	43.2 / 6.5	64.09	79.89
Varying α at $\kappa = 6$				
$\alpha = 1$	56.2 / 14.2	48.4 / 17.5	64.41	81.24
$\alpha = 2$	54.3 / 4.1	46.5 / 8.7	64.37	81.27
$\alpha = 4$	49.56 / 3.1	40.21 / 5.2	63.94	81.02

is capped. Therefore, a working episode scores below a silent one and the policy stops speaking. The sweep passes from drift to collapse with no interval in between at which the rate holds, which is why §2.2 treats what is missing as an attribution rather than a price.

Speaking-rate control Table 7 varies the two hyperparameters κ and α that the DLA introduces. Particularly, $\kappa = 0$ removes the channel and recovers GRPO exactly. All configurations start from the same SFT checkpoint and have no other differences.

The two hyperparameters act differently, as §2.2 predicts, and the κ block is what makes the difference visible. α moves the emission probability the policy settles at, tracking $\pi^\star = 1/(1+\alpha)$, and κ leaves that target untouched. The credit is not the only gradient reaching the gate, *i.e.*, the GRPO advantage pushes the rate up, since speaking more comes with the return of a good score, while the KL penalty pulls it towards the SFT rate. Both act relative to the credit and so weaken as κ grows, which predicts that the measured rate approaches $1/(1+\alpha)$ down the κ block. We present more detailed derivation and numeric analysis in Appendix B. What the benchmarks add is the cost of getting the rate wrong. For example, at $\kappa = 0$ the channel is absent and the duplicate rate on ProactiveVideoQA rises with the emission rate, whereas across the settings that hold a relatively flat OVO-Bench and StreamingBench, which is what a term acting only on the gate positions of proactive episodes should do to reactive capabilities.

C Staged supervision: group contrast and the order of the stages

This appendix supports §5.2 in two parts. It first derives the claim that a closed reasoning channel is an absorbing state rather than a slow one, and connects it to the entropy bonus of Equation 8; it then sets out what that analysis implies for the order of the two supervised stages.

The quantity. Write p for the probability that the policy opens a non-empty reasoning step at a given decision position, that is $p = 1 - \mathrm{softmax}(\ell_t)[\texttt{</think>}]$ from Equation 7, read at the token following <think>. A single forward pass measures it; no generation is required.

The estimator is comparative. GRPO draws G episodes per prompt, scores each with a scalar $R^{(i)}$, and standardizes within the group, so that $\hat{A}^{(i)} = (R^{(i)} - \mu_R)/\sigma_R$ and $\sum_i \hat{A}^{(i)} = 0$. The only proposition such an estimator can express is that one episode did better than its siblings; it has no baseline outside the group. Consequently, a binary decision can be informed only by a group in which both of its outcomes were actually sampled. If every episode in the group left the reasoning step empty, the group contains no evidence about the alternative: the gradient at that position is not identically zero, since the advantages differ for reasons elsewhere in the episode, but those contributions sum to zero over the group and carry no systematic preference between opening and closing the step.

The probability of a mixed group. Treating the G episodes as independent draws at the position,

$$P_{\text{mix}}(p) \;=\; 1 - p^G - (1-p)^G, \tag{11}$$

the probability that at least one episode opens the step and at least one closes it. Table 8 evaluates it.

Table 8: Fraction of reasoning decisions at which a group carries any contrast, from Equation 11. The function is symmetric about $p = \frac{1}{2}$ and vanishes at both ends.

p	0.017	0.05	0.10	0.30	0.50	0.90	0.99
$P_{\text{mix}}, G = 4$	6.6%	18.5%	34.4%	75.2%	87.5%	34.4%	3.9%
$P_{\text{mix}}, G = 8$	12.8%	33.7%	57.0%	94.2%	99.2%	57.0%	7.7%

Why the extremes absorb. Expanding Equation 11 about $p = 0$ gives $P_{\text{mix}}(p) = Gp + O(p^2)$, and about $p = 1$ gives $G(1-p) + O((1-p)^2)$. The learning signal available to correct a collapsing reasoning rate is therefore proportional to that rate. As the channel closes, the evidence that could reopen it disappears at the same order, and no restoring force remains; the same statement holds at the opposite end, where the evidence that the step could usefully be skipped disappears instead. This is what makes both extremes absorbing rather than merely slow to leave, and it predicts what our analysis of speaking-rate control observes, that an unregularized run departs to one extreme or the other, and that which extreme it reaches is not a property of the objective. The SFTKey model of Table 5 therefore offers almost no contrast: its reasoning rate is measured at 0.00, so a group of $G = 8$ effectively never samples both choices, against $P_{\text{mix}} = 98\%$ at the 59.93% our staged initialization leaves. The conclusion does not turn on the exact rates, since P_{mix} exceeds 0.9 at any rate between 0.25 and 0.75 and falls below 0.01 only below 0.0013.

Relation to the entropy bonus. P_{mix} is symmetric about $p = \frac{1}{2}$, vanishes at $p \in \{0, 1\}$ and is maximized in between, which is the shape of the binary entropy $H(p)$. At $G = 2$ the correspondence is exact,

$$P_{\text{mix}}(p) = 1 - p^2 - (1-p)^2 = 2p(1-p), \tag{12}$$

the Gini impurity, the standard quadratic surrogate for entropy; for larger G the two remain concave functions sharing their zeros, their symmetry and their maximizer. Equation 8 is therefore not only a device for keeping a behaviour alive. It maximizes, up to the choice of surrogate, the probability that the group-relative estimator has anything to compare at the position it acts on, which is why the supervised schedule of §2.1 and the regularizer of §2.2 are addressing one quantity from two directions.

What follows for the order of the stages. Training both losses at once keeps the channel open but is the weakest schedule on every track of Table 5. We attribute the direction to the structure of the sequence rather than to the data: an episode interleaves a reasoning step and a gate decision at each of T steps, so far more layout has to settle than in a single-span setting, and fitting trace content against a layout that is still moving is the cost the first stage removes. The same reading places the opposite order. Shi et al. (2025) supervise one reasoning span followed by one answer, where the answer is what is scored and devoting the final stage to it matches the objective; here the trace is written at every step and read by every later one, so it is the trace that the final stage has to secure.

Table 9: Curated corpora. A SOT video carries up to two episodes with different questions, so episodes exceed videos in that split. Reactive items carry one verifiable query, and proactive items carry an opening request and several reference answer turns, each valid only within a span of chunks.

	Split	Episodes	Answer turns	Source
SFT	SOT	68,351	2.17	LLaVA-Video-178K (Zhang et al., 2024), Tarsier2-Recap (Yuan et al., 2025)
RL	reactive	51,68	1	as above
	proactive	1,856	2.1	Live-WhisperX (Chen et al., 2025), Ego4D (Grauman et al., 2022), EgoExoLearn (Huang et al., 2024)

Assumptions. Equation 11 treats the G episodes as independent at the position in question. They share a prompt and diverge only as they sample, so the approximation is closest early in an episode and degrades as contexts separate. The rate p also varies across positions, and P_{mix} is concave in p, so by Jensen's inequality the average of P_{mix} over a heterogeneous set of positions is at most its value at the mean rate. Evaluating Equation 11 at a measured mean therefore overstates the contrast available, and the figures above are upper bounds on the signal a collapsed policy receives.

D DATA CURATION DETAILS

Curation process. Curation runs as three passes, ordered by what each requires of the others. Questions are generated first, because a trace is written against the answers it must later support evaluation of generated traces. Reply follows based on visual context. Traces last, whose quality is evaluated by the QA from previous steps, as mentioned in 3. Reasoning traces are produced by Qwen3-VL-235B-A22B-Thinking with thinking enabled, sampled at temperature 0.4 under decoding constrained to the output schema. Questions are generated by Kimi-K2.6 at 448px spanning the episode, conditioned on the record's response format and interaction mode, so that regeneration alters content while leaving the distribution over open-ended, multiple-choice, binary and counting items unchanged. Table 9 reports both corpora for SFT and RL. The two SFT splits differ in shape rather than only in provenance.

The trace curation prompt. The prompt below is the one used for every trace generation, reproduced verbatim. There is no system message: each call is a single user turn. The gate {`qline`} takes one of two forms: before the question is asked it reads *The viewer has not asked anything yet; you are simply watching*, and at or after the asking chunk it names the question and its time.

```
You are a streaming video assistant watching a video as it plays. You are currently at {now}
SECONDS. The frames above are what you have just seen, each labelled with its second.

{prior}{qline}=== ESTABLISHED FACTS ABOUT THIS VIDEO ===
{facts}
These are TRUE. You must never write anything that contradicts them. If they say something is
absent, never seen, or never happens, then it is NOT in this video and you must not describe
it as present -- not even tentatively. Do NOT restate these facts and do NOT reveal them as an
 answer; they only bound what you may claim.
========================================

Write a thought where you can pair something you actually SEE with what it TELLS you. Do not
work through the moments in order and do not aim to cover them. Most moments warrant nothing
and returning few thoughts, or none at all, is a correct answer.

FIRST job of each thought is to cover WHAT HAPPENED in this window -- what started, finished,
appeared, changed, or was handed over. Record the occurrence, not the scenery. Also CARRY THE
RUNNING STATE forward -- update it, never re-describe it.

RECORD THE VALUE, NOT THE PRESENCE. The observation half must name the specific thing you can
read off the frame -- WHICH object, WHICH colour, HOW MANY, WHAT the text says, WHICH step
number -- never merely that something is there.

THE TEST, which you can apply without knowing what anyone will ask: if your sentence would be
just as true of a DIFFERENT video of the same activity, it carries no value. Make it true of
THIS one.
```

```
BEFORE YOU WRITE, LOOK BACK AT THE RECORD ABOVE and ask what this moment connects to. Look
specifically for:
    - a RECURRENCE of something already recorded. Point back to WHICH earlier moment
    - the COMPLETION of something you recorded starting, or its abandonment
    - a REVERSAL of something recorded earlier
    - something that STILL has not happened, though enough time has passed that it should
    - a CHANGE from how the same thing looked when you last recorded it
When you find one, WRITE THE CONNECTION OR SOMETHING WORTH INFERRING. DO NOT conclude anything
 that rests on current moment alone. If you look back and find no connection, record the
occurrence and stop there. NEVER guess a value you cannot actually make out.

The state is small and fixed. Once the video scrolls out of view it is the ONLY thing you
still have, so it must be worth keeping:
    step progress through a procedure (3/6), if there is one
    entities what entered or left, marked + or - . At most 3, and only on the chunk where it
changed -- never re-list what is already being carried.

While writing the reasoning trace, add the state ONLY when it CHANGES -- a step advances,
something enters or leaves. When nothing in the state changed, write the reasoning alone if
available.

Rules:
- ABSENCE IS AN OBSERVATION. If something expected is not there, say so plainly and draw the
inference. This is correct and valuable.
- NEVER hedge about your own knowledge. Either you see something and say what it means, or you
 say plainly that it is not there, or you write nothing for that moment.
- ONLY what is visible in these frames and what you already thought. Never invent an object,
person or action to satisfy the question.
- ACCUMULATE, NEVER RE-LIST. Refer back, but never restate.
Output ONLY JSON: {{"thoughts":[{{"t":<second>,"think":"<reasoning | state>",}}, ...]}}
```

E TRAINING CONFIGURATION

Backbone and interface. All runs start from Qwen3-VL-4B-Instruct. `<silent>` and `<response>` are added to the tokenizer, where they occupy reserved embedding rows so that the embedding matrix is not resized; `<think>` and `</think>` are already present in the backbone. Sequences are cut at $32{,}768$ tokens throughout, which is what bounds the chunk count per episode.

Supervised stages. The two stages share an optimiser configuration — the full model including the visual encoder, batch size 8, no gradient accumulation — and differ only in the loss mask, which is set by a single flag. The first stage excludes the interior of every reasoning span from the loss; the second restores it, and resumes from the first at step 943 rather than starting afresh. Reinforcement learning initializes from the second stage at step 331. The first stage runs 943 steps at a peak learning rate of 10^{-5} after a 29-step linear warm-up, the second 331 steps at 5×10^{-6} after 10 steps of warm-up; both then decay on a cosine schedule.

Reinforcement learning. TSGPO runs on eight GPUs of a single node with two episodes per device, sampling $G = 8$ episodes per prompt, at a peak learning rate of 10^{-6} reached over a 58-step warm-up. Credit is assigned once per reply. The channel is configured with $\kappa = 6$, $\alpha = 2$ and a sign-scale cap $\tau = 6$. Both normalisations of §2.2 are active: the per-episode centring of Equation 4, and the running scale of Equation 5 with smoothing $\rho = 0.95$ and floor $\epsilon = 10^{-3}$. These last two are not exposed as flags but are the defaults of the normaliser the trainer constructs, so they hold whenever the running scale is enabled. The reasoning-rate entropy bonus uses $\beta_H = 3 \times 10^{-3}$, applied to the binary decision alone rather than the full vocabulary. Timing is scored in chunk-index rather than wall-clock space, which makes the reward robust to the per-video downsampling that sets Δ, and proactive turns are credited in recall mode.

Reward weights. On reactive episodes use the correctness reward and our proposed hidden-quiz reward, with $\omega = 0.5$. They are gated by the format checker. On proactive episodes, only answer correctness reward is used. We use it along with our proposed DLA to balance the correctness and repetitions.

Table 10: **Regularizing the reasoning decision.** One RL configuration, two treatments of $\hat{r}$. The unregularized run reaches the highest average by ceasing to reason; the rate it collapses *to* is set by the training mixture, not by the objective.

Regularizer	$\hat{r}$ (init $\rightarrow$ final)	RT	BT	FT	Avg.	RTVU
SFT	59.01	71.21	60.73	46.85	59.60	79.72
None	$59.01 \rightarrow 0.10$	71.55	54.70	44.83	57.03	79.92
Entropy bonus	$59.01 \rightarrow 59.93$	72.16	68.57	52.40	64.37	81.27

F REASONINGRATE REGULARIZATION

Left alone the reasoning rate does not stay where SFT put it. Table 10 follows $\hat{r}$ through an otherwise identical run under no regularizer and under the entropy bonus of Equation 8. For unregularized run, the rate leaves the interior within a few hundred steps and does not come back, and which extreme it reaches is a property of the training mixture rather than of the objective. Comparatively, the entropy bonus does not have to know the right rate. It has only to keep the decision from being settled, after which the task reward selects among rates that are still being sampled.

G UNPROMPTEDBENCH: ANNOTATION AND GROUND-TRUTH CONSTRUCTION

G.1 CANDIDATE DISCOVERY AND ANNOTATION INTERFACE

Candidate discovery is intentionally broader than the final answer key. A whole-video, category-balanced pass proposes a small set of procedural, safety, recommendation, and social interventions. A separate recall pass first constructs a log of events that actually occurred and proposes a call-back only when that remembered fact could help at the current moment. Additional safety-focused discovery reduces the risk that a sparse main pass overlooks rare but consequential events. These targeted prompts broaden the category mix and reduce unsupported recall proposals; they do not themselves certify grounding. Whole-video access is permissible at this discovery stage because proposals are only items for human review; it is never used to judge a model's causal output. Exact and semantic deduplication remove obvious proposal duplicates before review.

The annotation form decomposes a candidate into four questions rather than asking for a single accept/reject verdict:

1. **Relevance and grounding:** Is the message factually supported and useful, rather than a description of what is already visible?
2. **Timing:** Is it actionable at the proposed time, too early, too late, or dependent on future evidence? A timing failure permits a corrected timestamp.
3. **Redundancy:** Does it add information or action beyond earlier assistance under the evidence currently available?
4. **Egocentricity:** Does it preserve the camera wearer's perspective?

Separating these axes prevents a plausible sentence at the wrong time from being silently converted into a valid opportunity and makes the source of disagreement auditable. The v2 export pseudonymizes annotators but retains their identifiers: all 611 retained items have three ratings from three distinct annotator IDs, with 24 annotators overall.

G.2 AGREEMENT, ROUTING, AND EXPERT ADJUDICATION

The raw second-round pool contained 621 candidates in 49 videos. A pre-split privacy screen excluded 10 candidates and their one source video. The remaining 611 candidates in 48 videos have 1,833 ratings and split into 454 development candidates across 36 videos and 157 sealed-holdout candidates across 12 videos. This accounts for the full transition from proposal collection to the development catalog; it is not performance-conditioned filtering. The video-

Table 11: **Human agreement over all 611 retained second-round candidates.** Egocentricity is rare, so its high raw agreement and low κ must be interpreted together. These statistics diagnose annotation quality, not model performance.

Axis	Observed	Unanimous	Fleiss κ
Relevance	60.7%	42.2%	0.216
Timing	65.2%	51.6%	0.222
Redundancy	88.5%	82.8%	0.520
Egocentricity	96.6%	94.9%	0.145

disjoint 75/25 assignment is stratified by source family and human-outcome features with seed `sot-round2-v1-20260908`; the v2 manifest carries those video assignments forward unchanged. Development contains 15 EgoConv, 15 EgoLongQA, and 6 EgoProactive videos; holdout contains 5, 5, and 2, respectively. Table 11 reports observed agreement, unanimity, and Fleiss' κ by axis. The combination of moderate raw agreement on relevance and timing with much stronger agreement on redundancy makes a single undifferentiated "quality" label inappropriate. Egocentricity illustrates the complementary point: high raw agreement can coexist with low κ when violations are rare.

Timing review exposes ambiguity that aggregate agreement alone hides. At least one annotator flagged future leakage for 116/611 candidates (19.0%), but a majority did so for only 23/611 (3.8%). Reviewers supplied 154 rating-level timestamp corrections across 125 candidates; 121/154 (78.6%) moved the intervention later, with a median signed shift of $+9$ seconds. Three candidates carry an auxiliary `missing_relevance_reason` flag but have complete categorical labels and remain in the full analysis. Excluding only those three reproduces the earlier complete-rationale sensitivity (113/608 with any leakage vote; 153 corrections, 120 later; median $+9$ seconds) without changing the conclusion.

For an axis with label counts $n_{i\ell}$ among the three ratings of item i, reported observed agreement is the mean pair agreement $\bar{P} = N^{-1}\sum_i(\sum_\ell n_{i\ell}^2 - 3)/6$; unanimity is the fraction of items with one observed label; and Fleiss' $\kappa = (\bar{P} - \sum_\ell p_\ell^2)/(1 - \sum_\ell p_\ell^2)$, where p_ℓ is the corpus-level label prevalence. We use the native label sets: {relevant, invalid, unsure}, {timely, mistimed, leakage, unsure}, {nonredundant, redundant, unsure}, and a Boolean egocentricity flag.

For initial catalog status, each rater's holistic label is invalid if relevance is invalid, timing is mistimed or leakage, or egocentricity is violated; it is unsure if a remaining axis is unsure, and valid otherwise. A strict two-of-three majority is then applied. A majority leakage or majority egocentricity violation is rejected. Any single leakage or egocentricity flag is otherwise routed to adjudication. Majority-mistimed content is routed for retiming only when its non-timing content is valid and at least two raters provide corrected timestamps; otherwise it is rejected or adjudicated. Timely items with a majority-valid holistic label are accepted, majority-invalid items are rejected, and every residual case is adjudicated. During final resolution, the frozen majority-invalid policy rejected 64 already-clear non-retiming cases; one author adjudicator resolved the remaining 42 cases. This final pass is not an independent external expert panel, so we preserve every pre-adjudication rating and decision in the release for audit.

Table 12 gives the complete development-catalog accounting. The initial automated projection is replayed exactly and never overwritten. A frozen majority policy resolves 64 clear routed cases; the remaining 42 receive expert adjudication. The overlay records the before/after state, rationale, reviewer, timestamps, and content hashes for every decision. The output contains no unresolved or retime-required records.

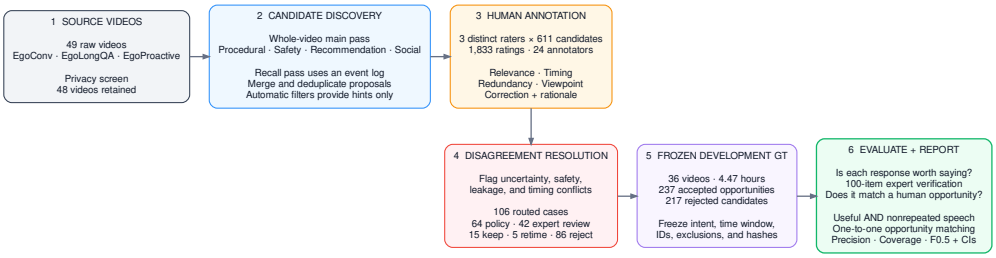


Figure 3: **UNPROMPTEDBENCH construction and evaluation.** Candidate discovery is separated from three-rater annotation, disagreement resolution, and freezing of the human ground truth. Evaluation then asks whether each response was worth saying and whether it addressed a human-identified opportunity.

Table 12: **Development catalog freeze accounting.** Every one of the 454 candidate records has a terminal status. "Routed outcome" partitions the 106 cases that were not initially terminal.

Stage / outcome	Count
Initial accepted / rejected	217 / 131
Adjudication-required / retime-required	98 / 8
Resolved by majority policy / expert adjudicator	64 / 42
Routed outcome: keep / retime / reject	15 / 5 / 86
Frozen accepted / rejected	**237 / 217**

Why expert review changes the benchmark. One procedural candidate advised, "Write the oxidation states under each element to track the changes clearly." All three reviewers found the content relevant and nonredundant, but timing split two-to-one because one reviewer flagged future leakage. A content-only majority would have accepted the original 3.0-second target. Expert review instead retained the useful intervention while marking the original point too early and moving its target to 18.5 seconds. The final catalog then constructs the same clipped ± 10-second acceptance window used for every accepted target. This example shows why content acceptance, causal timing, and catalog inclusion are separate decisions.

G.3 FROZEN OPPORTUNITY REPRESENTATION

Each accepted record stores a stable reference identifier, video identifier, category, human-approved response, target time, clipped acceptance-window endpoints, source record hash, and resolution provenance. The 237 accepted development opportunities are distributed as 126 procedural, 59 social/chit-chat, 20 safety, 17 recall, and 15 recommendation items across 32 of the 36 evaluated videos and 4.47 total video hours; four development videos contain no accepted opportunity. A clean replay from the frozen human inputs and expert-resolution overlay produced a byte-identical catalog and manifest; full content hashes are supplied with the release.

H UNPROMPTEDBENCH: EVALUATOR VERIFICATION AND SCORING

The evaluator has two sequential decisions. The *response-quality check* (formally, the event gate) asks whether a model utterance was worth saying at all. Only responses that pass are compared with the human catalog. The *opportunity-match check* (semantic matching) then asks whether a passing response delivers the same useful help as one particular human opportunity. The first decision determines precision; the second determines coverage.

Table 13: **Judge-visible expert verification.** One expert reviewer assessed every item. Positive-class metrics treat a response-quality pass and an opportunity match as positive. Semantic binary metrics apply the reported conservative policy (automatic `unsure`→non-match). Intervals are 95% video-cluster bootstrap intervals.

Stage	Agreement	κ	Precision	Recall	F1
Response quality	$42/50$ (84.0)	.643	60.0	100.0	75.0
Opportunity match	$47/50$ (94.0)	.860	100.0	82.4	90.3

Response-quality accuracy: 73.3–93.2; precision: 35.3–81.3; recall: 100.0–100.0; F1: 52.2–89.7. Semantic accuracy: 86.3–100.0; precision: 100.0–100.0; recall: 58.8–100.0; F1: 74.1–100.0. Values and intervals are percentages except κ.

H.1 Expert verification of judge decisions

Ground-truth opportunities and evaluator labels serve different purposes. The former define what should happen and when; the latter assess arbitrary model emissions and event–opportunity pairs. We therefore freeze a 100-item verification sample before expert review. Its 50 response-quality items comprise 20 automatic passes, 15 redundancy failures, 12 substantive validity failures, and 3 malformed outputs. Its 50 semantic pairs comprise 14 matches, 20 mismatches, and 16 unsure decisions and include every automatic match and unsure pair from the primary SoT checkpoint. The deterministic sample spans 28 videos and 11 result rows and is bound to the versioned evaluator matrix recorded in the release manifest.

One expert reviewer reviewed the causal video, candidate output, judge verdict and rationale, and linked three-rater or adjudicated opportunity evidence, then assigned an event validity/redundancy label or semantic match label. Because the automatic decision is visible, this is judge-visible expert verification: it is neither a blinded held-out test nor inter-annotator agreement. Table 13 reports the results. On the event gate, the judge agreed on 42/50 examples: all 12 expert-positive events passed, but 8/20 automatic passes were rejected by the expert, commonly because literal scene descriptions were not useful interventions. On semantic matching, all 14 automatic matches were confirmed; the conservative binary policy missed three of 17 expert matches. Exact three-way semantic agreement was 32/50 ($\kappa = .438$), because the expert resolved the 16 automatic `unsure` cases as 15 mismatches and one match. Collapsing `unsure` to non-match, as the scorer does, gives 47/50 agreement and $\kappa = .860$. These are stratified-audit diagnostics, not prevalence estimates.

The event pass precision (60.0%) and exact three-way semantic κ do not meet the prespecified verification thresholds. We therefore describe the evaluator as expert-verified but not fully qualified, retain conservative scoring, and expose the resulting uncertainty rather than tuning on these labels. Sample IDs, rubric, source hashes, and reporting thresholds were frozen before review; any revised judge requires a new versioned verification sample.

Evaluator selection is not based on which judge gives SoT the largest score. Natural contrast controls separately test wrong-object/action, temporal leakage, redundancy, social value, and malformed outputs. Candidate prompts or judge families that miss a quality, stability, provenance, or control gate remain sensitivity analyses and cannot define the headline metric.

H.2 Event validity and redundancy

For every emitted event the evaluator receives the utterance, same-video evidence up to the recorded cutoff, and all previous assistant utterances. It does not receive model identity or the human opportunity text. The validity decision requires causal visual grounding, factual correctness, current usefulness, safe advice, appropriate timing, and no reliance on future-only information. A retrospective pass may inspect later frames only to veto future leakage. Missing or malformed verdicts abstain and cannot earn credit. Each parsed user-facing utterance is scored as one event; a bundled response is valid only if every material claim and instruction passes the relevant criteria, preventing a system from hiding an unsupported claim inside an otherwise useful sentence.

Redundancy is semantic, not lexical. An emission is redundant when it provides no materially new information or action relative to earlier assistant speech under unchanged evidence. A reminder re-

Table 14: **Effect of redundancy-aware scoring on development precision.** Validity-only precision counts every event accepted by the validity stage; the final precision additionally bars redundant speech while retaining every emission in the denominator. R/V is the fraction of validity-eligible events removed as redundant. Asterisks mark tiny-denominator diagnostic rows; VST-7B uses the repaired, protocol-specific prompt-v4 adapter.

Model / checkpoint	Emissions	Valid	Final	Validity-only P	Final P	R/V
SoT checkpoint-700	447	293	214	65.55%	47.87%	26.96%
Qwen3-VL-4B parent	1,303	140	15	10.74%	1.15%	89.29%
Qwen2.5-VL-3B parent	14,397	6,181	37	42.93%	0.26%	99.40%
MMDuet2-3B	7,430	166	1	2.23%	0.01%	99.40%
Em-Garde	957	168	90	17.55%	9.40%	46.43%
VST-7B prompt-v4	231	98	18	42.42%	7.79%	81.63%
VST-32B*	2	1	1	50.00%	50.00%	0.00%
StreamOPD-4B*	14	7	4	50.00%	28.57%	42.86%
SoT checkpoint-700 (64K)	454	291	208	64.10%	45.81%	28.52%
SoT a278 RL-200 (64K)	206	154	99	74.76%	48.06%	35.71%
SoT c3082 RL-200 (64K)	1,218	167	120	13.71%	9.85%	28.14%
SoT SFT-332 (64K)*	2	2	2	100.00%	100.00%	0.00%

mains eligible when the situation has changed, risk has escalated, or a previously unavailable action becomes possible. Every emission stays in the precision denominator; only events passing both validity and nonredundancy enter the numerator or the coverage matcher. The frozen redundancy-only judge passes all 9 natural controls and reproduces all 9 labels in a separate repeat; this validates the redundancy subdecision but does not substitute for the 100-item expert verification of the complete evaluator.

H.3 Temporal and semantic matching

After the event gate, eligible edges are built only within a video and within the accepted opportunity window. The semantic matcher evaluates whether the model output realizes the same useful intent as the human reference. Mere object or action overlap is insufficient. Maximum-cardinality one-to-one matching is then recomputed from the surviving edges. This ordering matters: deleting a redundant emission from an old matching without rematching could discard an opportunity that another surviving event covers, whereas matching before gating could award coverage to an ineligible repeat.

Matcher-boundary sensitivity. The primary policy treats an `unsure` semantic verdict as a mismatch. On the completed 237-opportunity development matrix, checkpoint-700 has 4 conservative matches and $F_{0.5} = 7.40\%$. Counting `unsure` as a match raises these to 9 and 14.41%, respectively, while leaving its $214/447 = 47.87\%$ precision unchanged. This large boundary sensitivity is why the conservative rule is fixed as primary and the inclusive value is not a competing headline score. Expert verification supports that choice: all six `unsure` pairs from primary checkpoint-700 were labeled mismatch. Across the full 50-pair audit, however, two automatic mismatches were judged matches, so we separately report a human-correction sensitivity without replacing the automatic headline matrix.

The scoring output for each system therefore retains:

- total emissions and emitted interventions per hour;
- valid-and-nonredundant numerator, precision, and confidence interval;
- matched opportunities, the fixed opportunity denominator, coverage, and confidence interval;
- $F_{0.5}$ and its source-family-stratified video-bootstrap interval;
- redundancy and any populated mistiming, future-leakage, malformed-output, and no-emission diagnostics (missing fields remain unavailable, not zero); and

- protocol tier, adapter status, checkpoint identity, and artifact hashes.

H.4 MODEL-SIDE PROTOCOL CONTROLS

The saved development outputs were selected using a five-video panel containing three positive and two zero-opportunity controls. The locked configuration retains 20 visual chunks, permits 20 thought tokens and 60 new tokens, uses greedy decoding with seed 0, and supplies the fixed session-level instruction "Tell me what to do next." This is not a turn-level query identifying a target opportunity. Candidate configurations had to pass anti-silence, anti-spam, category-breadth, timing, leakage, and redundancy checks. The benchmark holdout and judge-audit fold were not used for this model-side selection.

Released systems differ in whether they natively support query-free streaming. Accordingly, we report native query-free systems, query-free adaptations, generic-query controls, and prompted controls as separate protocol tiers. Raw-output audits verify completeness, decision cadence, causal frame timestamps, and parsing. An inaccessible checkpoint is "unavailable," not zero; a broken adapter is "adapter failure," not evidence of model incapability; and emission-conditioned precision is undefined for a silent system, for which coverage and $F_{0.5}$ are zero by convention.

H.5 FROZEN MODEL AND EVALUATOR PROMPTS

The session-level model request is exactly: *"Tell me what to do next."* (prompt P0). The locked C0 configuration uses 20 visual chunks, 20 thinking tokens, 60 maximum new tokens, greedy decoding, and seed 0. Query-free parent-model adapters additionally receive the system message *"You are an egocentric assistant watching a live video stream. Use only visual evidence observed up to the current moment."* At each decision, the adapter lists up to 20 prior utterances and appends: *"Think privately inside* `<think>` *and* `</think>`*. After* `</think>`*, output exactly* `NO_REPLY` *if silence is best; otherwise output exactly one brief sentence for the wearer. Do not expose private reasoning or output any other markup."* Native and adapted interfaces are reported as different protocol tiers.

The evaluator uses Qwen3.6-27B with deterministic decoding and structured JSON. Below we reproduce the invariant instructions verbatim in italics; only the per-item JSON and output-key boilerplate are omitted.

Response-quality judge (formal event-validity stage).

> *"You are applying the frozen SoT Round 2 human annotation rubric. Judge the candidate at its emission time using ONLY causal evidence. You have not been shown any frames after the emission cutoff. Do not label future leakage in this pass; use unsure when the timing cannot be established from the causal evidence. Score each dimension independently. Relevant means correct, grounded, and useful to the wearer. Judge relevance independently from exact timing. Redundancy verdict is a prior-speech duplication diagnostic only; it cannot affect validity."*

The same versioned source contract first asks an evidence-only model to record visible observations, supported, unsupported, and uncertain claims without deciding whether the assistant should speak.

Future-timing check.

> *"You are diagnosing post-cutoff timing errors in a proactive response. The causal evidence was available when the response was emitted. The post-cutoff evidence occurred later. Independently diagnose timing from this evidence; no causal-pass verdict is provided. Score timing only. Never change or score relevance, usefulness, redundancy, egocentricity, or overall validity."*

This pass may identify leakage or the first acceptable later time, but it cannot upgrade a response that failed using causal evidence.

Redundancy judge.

> *"You are applying the frozen Unprompted Bench redundancy rubric. Judge only whether the current unsolicited assistant utterance adds enough new user value to justify another interruption. Use only evidence at or before the current emission time. Do not judge correctness, relevance, timing, leakage, style, or overall validity. Return redundant only when a prior assistant utterance already communicated the same practical or social intent and the current utterance adds no materially new information or action under substantially unchanged evidence. Paraphrases and repeated generic narration are redundant. A changed visual state alone is insufficient; it must make another unsolicited interruption useful or necessary."*

Its exact source is identified in the release manifest.

Opportunity-match judge (formal semantic matcher).

> *"Decide whether CANDIDATE conveys the same useful intervention as REFERENCE. First identify the CORE of each message: the action, warning, reminder, or fact that changes what the user should do or know now. Treat rationale, reassurance, politeness, urgency, and minor procedural detail as SUPPORT unless they change the required action, outcome, or safety constraint. Use MATCH when the CORE is equivalent. Allow harmless omissions of SUPPORT, ordinary differences in action granularity, and declarative-versus- imperative phrasing. Use MISMATCH when the action, object, direction, outcome, fact, or safety constraint differs, or when CANDIDATE merely mentions the same scene or object. Use UNSURE only when the CORE may be implicit and reasonable experts could disagree."*

The conservative scorer maps `unsure` to non-match. The exact prompt contract is identified in the release manifest.

All inputs are quoted as data, model identity is omitted, and the output schemas admit only enumerated labels, confidence, and a bounded rationale. The released evaluation package includes the complete verbatim templates, JSON schemas, model fingerprint, and per-item instantiated-prompt hashes; this summary is not used to regenerate scores.

H.6 Uncertainty and reproducibility contract

Confidence intervals are computed by resampling videos within source family and using the same resampled sessions for every compared system. The primary interval captures video-sampling uncertainty conditional on the frozen human catalog and evaluator. Judge-policy, repeated-run, temporal-window, and label-policy sensitivities are reported separately because they represent different uncertainty sources.

The release binds the split, opportunity catalog, model outputs, normalized events, judge prompts and identities, raw and parsed votes, post-gate match graph, bootstrap seed and replicates, final matrix, and generated LaTeX to SHA-256 hashes. A deterministic script checks denominators and recomputes every displayed percentage from integer counts; hand-edited result cells are not part of the paper workflow.

I Qualitative Streaming Trajectory

I.1 What the benchmark rewards and rejects

Tables 15 and 16 expose the evidence, model output, human context, and judge decision. They include successful matches from the primary model and its parent alongside failures from baseline and sensitivity configurations, demonstrating that the same rules can both grant and withhold credit. They were selected after inspection for explanatory value and are therefore illustrative, not a performance sample.

Table 15: **Responses correctly credited by the evaluator.** The same scoring rule recognizes useful interventions from the proposed model and a baseline. Examples are explanatory rather than a performance sample.

Frame	Model	Generated response	Human annotation context	Judge result
	SoT checkpoint-700	"Interact with the touchscreen displaying 'HUMAN HAMSTER WHEEL' beside the red 'lifestyle is a powerful' sign."	"Tap the screen to start the tunnel—it won't move until you trigger it, so no rush." Window: 52–72 s. Human vote: 3/3 valid, timely, nonredundant.	Valid, nonredundant, and `match` (0.90). Earns precision and coverage.
	Qwen3-VL-4B	"Touch the screen to start the Human Hamster Wheel exhibit."	Same unanimously accepted opportunity and 52–72 s window as above.	Valid, nonredundant, and `match` (0.90). The evaluator credits the baseline under the same rule.
	SoT checkpoint-700 (64K)	"Unplug the black cord from the white wall outlet beside the bread machine."	"Unplug the appliance when you're done—it's a good habit to avoid accidental startups." Window: 107–127 s. The final expert resolution accepted the opportunity after the initial three ratings disagreed.	Valid, nonredundant, and `match` (0.95). The specific wording preserves the same safety action.

I.2 THINK–SILENT–SPEAK TRAJECTORY

The qualitative example is generated from an exact saved development trace only after joining it to the frozen judgments and post-gate match. We show source frames, the human opportunity and window, silence before sufficient evidence, a concise author-written summary of the model state, the exact emitted intervention, subsequent silence when the evidence has not changed, and each evaluator verdict. We do not publish hidden chain-of-thought text; any state description is a faithful summary of an available trace and is labeled as such.

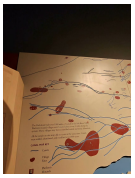


326.53 s: `<silent>` 329.53 s: `<silent>` 330.53 s: `<response>` 332.53 s: `<silent>`

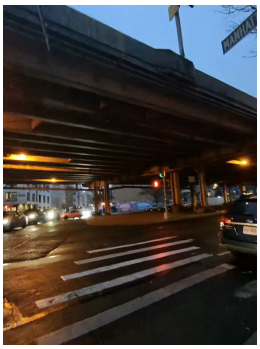
Figure 4: **Exact checkpoint-700 silent–speak–silent trajectory.** The model remains silent while food preparation continues and as the wearer approaches the sink, speaks once as the hands enter the running water, then returns to silence. Frames and gates come from the saved causal development rollout.

The human safety opportunity is "Rinse your hands after handling avocado—it is slippery and can make the knife hard to grip," with target 330.0 seconds and frozen window $[320, 340]$. At 330.53 seconds, the model emits exactly, "Turn on the faucet at the kitchen sink to wash your hands under

Table 16: **Responses correctly rejected by the evaluator.** Correct timing or visual overlap is insufficient when the response is stale, repetitive, or does not deliver the useful human intervention.

Frame	Model	Generated response	Human annotation context	Judge result
	StreamOPD-4B	"The Holiakam built over 1000 miles of canals in the Sal River Valley." Earlier outputs repeatedly corrupt the same exhibit text.	"Follow the blue lines from the river outward—they show how water was distributed to different settlements." Human vote: 3/3 valid, timely, nonredundant.	**Rejected:** invalid and redundant (0.95 redundancy confidence). No precision or coverage credit.
	MMDuet2-3B	"A person is holding a smartphone, seemingly taking a photo of the festive display, with a decorative display featuring a large wheel surrounded by colorful flowers, intricate architectural models, a small statue near the wheel, a decorative display featuring a small statue near the wheel, a decorative display featuring a small statue near the wheel, a decorative"	A nearby accepted opportunity directs the wearer to follow the train's red cars around the circuit. Human vote: 2/3 valid; 3/3 nonredundant.	**Rejected:** invalid and redundant; the rationale cites identical outputs at 240–242 s and no new decision point.
	SoT (64K)	"Plug the black cord into the white wall outlet." The model had already issued the same instruction at 81.53 s, and the cord was plugged in by 85.53 s.	"Make sure the cord isn't tangled before plugging it in—prevents strain and potential hazards." Target: 83 s. Human vote: 3/3 valid, timely, nonredundant.	**Rejected:** invalid and redundant (0.95 redundancy confidence). The stale instruction earns no credit; the human target adds a new safety constraint.

running water." Its saved thought field is empty at these four decisions, so we do not invent a rationale. The event passes the frozen validity and nonredundancy gates; the latter assigns `new_action` because no prior assistant utterance mentioned the sink or hand washing. The final semantic matcher labels it a match to the human intent (confidence 0.90). It therefore contributes once to the precision numerator and covers exactly one opportunity.



450.53 s

Thinking without speaking. In a separate exact checkpoint-700 development step, the internal field is nonempty while the user-visible gate remains `<silent>`. An author summary of the saved state is: *the Manhattan Street sign is visible and the wearer is approaching the intersection*. No accepted human opportunity is active at that timestamp. This example illustrates that updating stream state need not force an interruption; it does not expose the private trace verbatim.

Figure 5: **A nonempty state update with a silent user-facing gate.** The image is the causal source frame associated with the saved decision.

J EVALUATION LIMITATIONS

The current result matrix is development-only: 36 videos and 4.47 hours are too small for a claim of broad real-world coverage. The opportunity catalog is finite, and candidate-discovery recall has not been estimated by an independent exhaustive human sweep. Coverage therefore means coverage of the frozen human-validated catalog, not recall over every conceivable useful intervention; reference-blind precision partly addresses this but inherits judge error. Categories are imbalanced, with only 15 recommendation and 17 recall opportunities. Visual-only evaluation omits audio cues and therefore does not measure all reasons a wearable assistant might speak. The uniform clipped ± 10-second window is an operational tolerance around an adjudicated target, not a claim that every intervention has the same temporal utility profile. Adapted baselines test usable released interfaces rather than a pure architecture comparison, and failed or inaccessible adapters cannot support rank claims. Finally, bootstrap intervals condition on the chosen catalog and evaluator; they do not erase annotation disagreement or judge-policy uncertainty. We therefore report both sources explicitly and reserve held-out or state-of-the-art claims until a sealed evaluation is completed.