SELF-FOCUS: NATIVE ACTIVE PERCEPTION FOR STREAMING INTERACTION MODELS

Anonymous authors
Paper under double-blind review

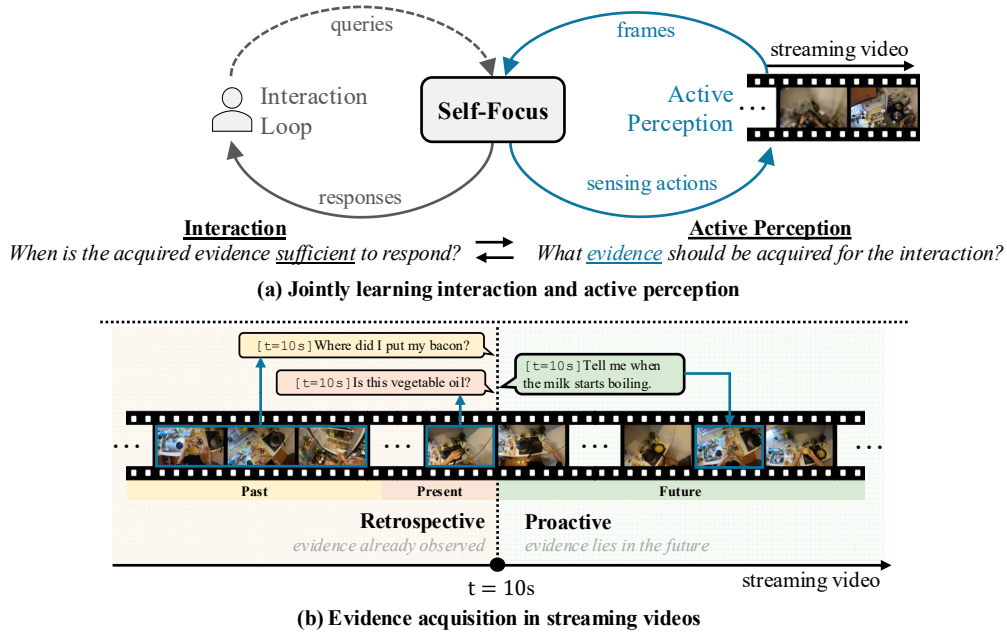


Figure 1: **Self-Focus jointly learns interaction and active perception.** (a) A single policy controls what evidence to acquire and when the acquired evidence is sufficient to respond. (b) In streaming video, queries may depend on evidence already observed (retrospective) or yet to arrive (proactive). The model must preserve or recover relevant history while acquiring new evidence as events unfold.

ABSTRACT

Deciding when to respond depends on the evidence a model has acquired. We introduce **Self-Focus**, a streaming interaction model that learns active perception natively within its interaction policy; thereby jointly deciding what evidence to acquire and when it is sufficient to respond. As the video stream progresses, Self-Focus constructs its own working context through active sensing, continuously, before and after a query arrives. We initialize the policy through supervised fine-tuning on our streaming episodes, then optimize it with GRPO over the policy's self-generated context. We introduce a *delayed sensing credit* to estimate each sensing action's advantage in subsequent interaction return. Self-Focus achieves state-of-the-art performance on both proactive and retrospective tasks across four benchmarks. The resulting query-agnostic context also improves offline long-video reasoning and provides a stronger initialization for iterative video agents, enabling better performance at fewer search iterations.

1 INTRODUCTION

Video-language models typically operate through turn-based interactions: a user provides a video and asks a question, and the model responds (Li et al., 2024; Bai et al., 2025). Streaming video changes this setting: observations arrive continuously, and events unfolding in real time may call for a response without waiting for a user query (Chen et al., 2024; Chatterjee et al., 2025; Xu et al., 2025a). A cooking assistant, for example, could proactively warn when milk is about to boil over. Recent interaction models (Thinking Machines, 2026; Cui et al., 2026; Yao et al., 2026) learn to

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.

take such initiative by deciding when to respond or remain silent (Fig. 1a). However, this decision depends on a more fundamental question: has the model acquired *sufficient* evidence to respond?

Existing interaction models focus on learning when to respond or remain silent, while relying on separately designed procedures for evidence acquisition. These procedures, including uniform frame sampling (Wang et al., 2025a), rolling KV caches (Xu et al., 2025a), and long-term summarization (Chatterjee et al., 2025; Yao et al., 2026), primarily aim to make arbitrarily long streams manageable within a bounded working context, rather than optimizing evidence acquisition for interaction. But deciding when to respond depends on what the model has seen, and deciding what to observe depends on what it may need to answer. This reciprocal dependency motivates jointly learning interaction and active perception within a single policy.

Unlike classical active perception (Bajcsy, 1988; Aloimonos et al., 1988), which often controls a sensor's configuration or viewpoint to acquire more informative observations, active perception in streaming video operates over an *exogenous* stream: observations arrive independently of the model's actions. The model instead controls which observations to retain and which to examine at higher fidelity, actively generating its own working context. Consequently, although the incoming stream evolves independently of the model, the evidence available for subsequent sensing and interaction decisions depends on its earlier actions.

Existing active perception methods largely study offline long VideoQA, where a known query guides evidence acquisition from an entire video through iterative temporal retrieval (Ding et al., 2026; Xing et al., 2026; Wang et al., 2026b) or adaptive visual processing (Zhu et al., 2026; Lee et al., 2026). Streaming video removes this privilege: sensing decisions need to be made as frames arrive, often before knowing what will be asked. A retrospective query such as *"Where did I put my bacon?"* requires preserving or recovering evidence from the past, while a proactive query like *"Tell me when the milk starts boiling"* requires continued sensing until sufficient evidence arrives (Fig. 1b); some queries require both. The model should therefore preserve keyframes before a query arrives and adapt its sensing once it does, all within a bounded working context. What is worth retaining, however, may only become clear in a future interaction.

We introduce **Self-Focus**, a streaming interaction model that learns active perception natively within its interaction policy. The policy controls what evidence to acquire and when that evidence is sufficient to respond. Each frame first arrives as a low-resolution preview; the model can `drop()` it, `keep()` it, or `focus()` to acquire a higher-resolution observation. Once a query arrives, it may also `retrieve()` an interval from the observed past. The policy then decides whether to emit a `<response>` or remain `<silent>`. These sensing decisions construct a *self-generated context*: the evidence the model has chosen to acquire and retain. Before a query arrives, the policy builds this context query-agnostically; once one arrives, it adapts acquisition to the active query.

To this end, we construct streaming episodes from temporal grounding datasets (Guo et al., 2025; Lei et al., 2021; Shao et al., 2020; Bärmann & Waibel, 2022) and cold-start Self-Focus through single-step supervised fine-tuning (SFT), thereby teaching the sensing and response actions. Temporal annotations tell us where relevant evidence lies, but not which sequence of sensing decisions will best support a future interaction. Such supervision therefore does not optimize how the model's own sensing decisions shape the context available for future interactions. Thereby, we train with GRPO (Shao et al., 2024) over causal streaming rollouts from a shared prefix. The rollouts observe the same future frames and queries, but construct different working contexts through their own sensing actions. An *interaction reward* evaluates each rollout for answer correctness, response timing, valid interaction, and context use. Yet this reward does not identify which earlier sensing decisions contributed to the outcome. To address this, we introduce *delayed sensing credit*, which estimates each action's advantage in future interaction return relative to the other rollouts. In this way, future interaction outcomes jointly optimize sensing and response actions, training evidence acquisition even before a query is known. Training across three context budgets (low/medium/high) further allows the policy to adapt its sensing to the available capacity.

We evaluate Self-Focus on proactive and retrospective Online VideoQA, testing its ability to acquire evidence as events unfold while recovering evidence from the observed past. It establishes state-of-the-art performance on SPOT-Bench (Chatterjee et al., 2026), ProactiveVideoQA (Wang et al., 2025b), and StreamingBench-PO (Lin et al., 2024), while advancing the perception-memory frontier on OVO-Bench Niu et al. (2025), achieving the strongest backward-memory performance alongside the best real-time accuracy. Controlled ablations further demonstrate the value of learning active perception within the interaction policy: jointly optimizing sensing and response actions improves

performance across proactive, real-time, and backward tasks compared with learning interaction alone.

We then examine whether the self-generated context remains useful beyond streaming interaction. A single query-agnostic scan produces reusable context that outperforms the base Qwen3-VL and the closest interaction model, JoyAI-VL-Interaction (Yao et al., 2026), on offline Long VideoQA (Fu et al., 2025; Mangalam et al., 2023; Zhou et al., 2025; Wu et al., 2024). We further use this context to initialize frozen long-video agents, replacing their usual starting observations without modifying their policies. Across different agents, this improves answer accuracy while reducing subsequent search iterations, demonstrating that the acquired context provides a stronger initialization for downstream evidence seeking.

To summarize: *(i)* We propose Self-Focus, which learns active perception within a streaming interaction policy. *(ii)* We introduce delayed sensing credit to learn query-agnostic evidence acquisition from future interaction outcomes. *(iii)* We advance proactive and retrospective Online VideoQA, with self-generated context that also improves offline long-video reasoning.

2 Related Work

Streaming Interactions. Streaming video enables mixed-initiative interaction between a user and a model. Online VideoQA benchmarks, where each query carries an ask time, have made this setting measurable. Early benchmarks predominantly evaluated retrospective queries (Niu et al., 2025; Lin et al., 2024). To maintain access to the growing stream, prior work have proposed fixed-rate sampling (Xia et al., 2026; Yao et al., 2025; Xu et al., 2025a; Chen et al., 2025), efficient KV caches (Di et al., 2025; Zhang et al., 2026; Pang et al., 2026), or textual summaries (Chatterjee et al., 2025; Guan et al., 2026; Liu et al., 2026). These models, however, primarily respond to user-initiated queries. More recently, proactive benchmarks (Chatterjee et al., 2026; Wang et al., 2025b) introduced another axis of evaluation: response timeliness. This has motivated *interaction models* that natively learn when to respond or remain silent (Wang et al., 2025a; Cui et al., 2026; Yao et al., 2026; Thinking Machines, 2026). Such response decisions depend on whether the model's context holds sufficient evidence. Yet such evidence acquisition and retention remain separately designed rather than learned within the interaction policy.

Active Perception in Video Understanding. Active perception treats sensing as a decision (Bajcsy, 1988; Aloimonos et al., 1988). Early work focused on learning frame-selection (Wu et al., 2019a;b) and stopping policies (Ghodrati et al., 2021) for action recognition. Recent video MLLMs employ query-aware keyframe selection and learned selection policies (Tang et al., 2025; Zhang et al., 2025; Zhu et al., 2026; Xu et al., 2025b; Qin et al., 2026), query-agnostic compression (Islam et al., 2025; Wang et al., 2026a), or iterative search (Cao et al., 2025; Ding et al., 2026; Zuo et al., 2026; Yang et al., 2025; Xing et al., 2026). These approaches establish the value of active perception, but operate primarily on the full video with known queries. Streaming instead requires causal evidence acquisition before a query is known, followed by query-conditioned sensing and possible recovery of the observed past. Self-Focus learns these decisions by jointly optimizing sensing and response actions through downstream interaction outcomes.

3 Self-Focus

3.1 Preliminaries: Streaming Interactions

A video stream delivers frames $v_1, v_2, \ldots$ at a fixed fps, where v_t becomes observable only at time t. The stream is exogenous and progresses independently of the model's actions. A user may pose a query q at any time τ_q, associated with an evidence interval $[s_q, e_q]$ that is not revealed to the model. The query's position relative to τ_q defines the interaction type (Fig. 1b): retrospective ($\tau_q \geq e_q$), including queries about the present ($\tau_q \approx e_q$); proactive ($\tau_q < s_q$), whose evidence has yet to arrive; and mixed ($s_q \leq \tau_q < e_q$), whose evidence spans query arrival. Before a query arrives, sensing is necessarily query-agnostic; after it arrives, sensing becomes query-conditioned. The former requires retaining potentially useful evidence before its utility for a future query is known. Under a bounded context, preserving such past evidence competes with representing the present, giving rise to a perception-memory trade-off (Shen et al., 2026; Chen et al., 2026).

```
<|im_start|>system
<system prompt>\nContext budget:  medium.<|im_end|>
<|im_start|>user
[stream]
[query @ <96.0 seconds>] Where did I put my bacon?
[answered @ <96.0 seconds>] In the fridge.
<104.0 seconds><image>                                                    keep()   <silent>
<108.0 seconds><image>                                                    keep()   <silent>
          ⋮
~~<114.0 seconds><image>~~                                                drop()   <silent>
~~<115.0 seconds><image>~~                                                drop()   <silent>
[query @ <116.0 seconds>] Tell me when the milk starts boiling.          retrieve(104,108)
[retrieved <104.0 seconds>-<108.0 seconds>]
<104.0 seconds><image> ...  <107.0 seconds><image>                                 <silent>
<116.0 seconds><image>                                                    keep()   <silent>
<117.0 seconds><image>                                                   focus()   <silent>
<118.0 seconds><image>                                                   focus()   <silent>
<119.0 seconds><image><|im_end|>
<|im_start|>assistant keep() & <response> The milk has started
boiling.<|im_end|>
```

Figure 2: **Streaming interaction grammar.** An example at $t = 119$ s, where the model responds to a query at $t = 116$ s. Frames, queries, and previous responses form a single temporally ordered user context. The right margin shows the sensing and response actions taken at each step; these actions are executed but are not retained in the subsequent conversation history. Struck-through frames are dropped from the working context, while `<image>` denotes a frame's visual-token block.

3.2 Streaming Interaction Grammar

Standard MLLMs (Bai et al., 2025) assume turn-based interactions, where a query follows the observed video and immediately elicits a response. In streaming interactions, the model instead decides both *when* sufficient evidence is available to respond and *what* evidence to acquire and retain along the way. Self-Focus makes sensing and response actions part of a single interaction policy, operating over a temporally ordered user context (Fig. 2). Video observations retain Qwen3-VL's native timestamp tags, while queries and previous responses are interleaved at their respective timestamps as `[query @ <t>]` and `[answered @ <t>]`. This preserves dialogue history without starting a new conversational turn at every timestep.

Sensing actions. Sensing actions determine which observations enter the model's working context, bounded by a context budget B specified in the system prompt Sec. E. Every incoming frame is first presented as a low-resolution preview of N_L tokens. From this preview, the model selects `drop()` to discard it, `keep()` to retain it, or `focus()` to acquire and retain a higher-resolution observation of N_H tokens. When the context exceeds B, the oldest retained frames are evicted, so acquiring new evidence can displace observations from the past. When a query arrives, the model may also call `retrieve(`t_s`,`t_e`)` to recover a contiguous interval from the observed past.

Response actions. At every timestep, after sensing, the model emits either `<silent>` or `<response>` followed by an answer. Response timing is therefore a native model decision rather than being controlled by an external harness (Chatterjee et al., 2026). The same policy thus determines both what evidence to acquire and when sufficient evidence is available to respond.

3.3 Constructing Streaming Episodes

We construct streaming episodes from temporal grounding datasets (Lei et al., 2021; Gao et al., 2017; Anne Hendricks et al., 2017; Bärmann & Waibel, 2022; Shao et al., 2020; Guo et al., 2025), which associate captions or queries with evidence intervals. By varying when a query arrives relative to its evidence, we create three types of interactions. For short events ($\leq$ 5s), queries precede the evidence, creating proactive interactions where the model must wait for the event to unfold. For longer events ($>$ 5s), queries arrive either after the evidence (retrospective) or within its final 5s (mixed), the latter requiring recovery of past observations and continued sensing. We further augment retrospective interactions with video-level QA from offline datasets (Zhang et al., 2024), placing queries at the end of the clip without requiring temporal annotations. Queries from the same video are combined into multi-turn episodes, retaining earlier queries and responses as dialogue history. In total, we construct 100K episodes containing 40% proactive, 30% retrospective, and 30% mixed interactions. These episodes provide query arrival times and response targets for both training stages. Further details are provided in Appendix D.1.

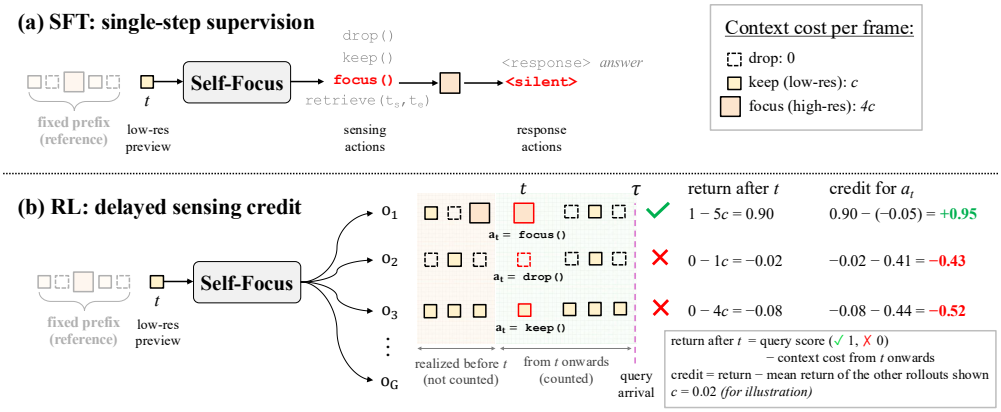


Figure 3: **Training Self-Focus.** **(a)** SFT replays a reference prefix and supervises one sensing action and one response action at step t, here `focus()` followed by `<silent>`. **(b)** RL samples G rollouts from a shared prefix: they receive the same frames and the same query, which arrives at τ, but take different sensing actions and hence hold different working contexts. The action a_t, taken before the query is known, is credited by its rollout's return after t, *i.e.* the query score minus the context cost from t on, relative to the mean return of the other rollouts (Eq. (2)); only o_1, which focused on the frame the query later needs, earns positive credit.

3.4 Supervised Fine-Tuning

Supervised fine-tuning (SFT) teaches Self-Focus the streaming interaction grammar (Sec. 3.2) and initializes its sensing and response behavior. Since evidence annotations do not specify a unique sensing trajectory, we derive reference sensing trajectories using simple, budget-conditioned rules: sparse `keep()` observations during dead-time and around past evidence, denser `keep()` and `focus()` when relevant evidence is observed, `retrieve()` when past evidence is needed, and `<response>` within the target response window. Rather than supervising complete episodes, which are dominated by `drop()` and `<silent>`, we sample individual decision steps and replay the preceding reference actions to reconstruct the working context and dialogue history under the sampled context budget. Each instance supervises a sensing action followed by a response action. We balance the sampled action classes and apply class-balanced cross-entropy to action tokens alongside the standard language-modeling loss. SFT provides the initial policy for RL, which then learns from the interaction outcomes of its own sensing and response decisions.

3.5 Reinforcement Learning

Streaming rollouts. SFT initializes the policy from reference actions, but does not optimize for the self-generated context. We therefore train the SFT checkpoint with GRPO (Shao et al., 2024) on self-generated rollouts, optimizing the resulting interaction as a whole. For each group, we sample a segment of an episode, replay the preceding reference actions to establish a shared prefix, and generate G rollouts $\{o_1, \ldots, o_G\}$ from it (Fig. 3b). Queries are revealed only upon arrival, while earlier queries and responses remain in the dialogue history. All rollouts share the incoming frames, query arrival times, and context budget, but their sampled actions produce different working contexts and subsequent decisions.

Interaction reward. Each rollout receives a trajectory-level interaction reward:

$$R = \lambda_{\text{fmt}} R_{\text{fmt}} + \lambda_{\text{task}} R_{\text{task}} + \lambda_{\text{time}} R_{\text{T-F1}} - \lambda_{\text{ctx}} C_{\text{ctx}}. \tag{1}$$

Here, R_{task} scores answer correctness, while $R_{\text{T-F1}}$ uses Timeliness-F1@1 from SPOT-Bench (Chatterjee et al., 2026), counting a response as a true positive only when it is correct and falls within the target response window; unmatched windows become false negatives. R_{fmt} penalizes violations of the interaction grammar, and C_{ctx} measures visual tokens committed through `keep()`, `focus()`, and `retrieve()`, normalized by the context budget B. Consequently, acquiring the same number of tokens incurs a greater cost under a smaller budget. We compute the reward terms per query and sum them over the segment (Appendix D.3). While this reward evaluates the overall interaction, it does not attribute its outcome to individual sensing and response decisions.

Delayed sensing credit. Much of a stream consists of dead-time, when no query is active. A frame retained during this period may only prove useful when a query arrives minutes later, after a series of sensing-response actions. Standard GRPO assigns every action in a rollout the same trajectory-level advantage. Ideally, each action should instead be credited by its advantage in subsequent interaction return, $A_t = \mathbb{E}_\pi[R_{>t} \mid h_t, a_t] - \mathbb{E}_\pi[R_{>t} \mid h_t]$, where h_t comprises the current observation, working context, dialogue history, and context budget. $R_{>t}$ is the portion of R not yet realized at step t *i.e.* excluding scores from queries whose response windows have passed and costs of tokens committed before t. Once a query arrives, sensing can be conditioned on it; before then, the model does not know whether the current observation will be useful for a future query.

Group-relative credit estimation. Estimating A_t directly could involve a learned critic to predict future return, requiring an additional model, or counterfactual rollouts that continue the stream from each decision step under alternative actions. For a segment of T decision steps and $|\mathcal{A}|$ sensing actions, the latter requires $O(|\mathcal{A}|T)$ continuations per rollout and $O(|\mathcal{A}|T^2)$ simulated steps, compared with $O(T)$ for the original rollout, amounting to thousands of continuations per group.

We exploit the exogenous stream: sensing changes each rollout's working context, while future frames and queries remain shared (Mao et al., 2019). We credit each action by its future interaction return relative to other rollouts, using a leave-one-out reference (Ahmadian et al., 2024):

$$\hat{A}_{i,t} = \underbrace{R^i_{>t}}_{\text{rollout } o_i} - \underbrace{\frac{1}{G-1}\sum_{j\neq i} R^j_{>t}}_{\text{other rollouts, shared future}} . \tag{2}$$

A positive $\hat{A}_{i,t}$ indicates that rollout o_i achieves a higher future interaction return than the other rollouts on average. When retained evidence improves a future interaction, that outcome can therefore credit earlier sensing decisions, including those taken during dead-time before the query was known (Fig. 3b). Although working contexts diverge after the shared prefix, $\hat{A}_{i,t}$ is not a same-state estimate of A_t; however, the leave-one-out reference depends only on other rollouts, preserving an unbiased policy gradient (Appendix D.4). We use $\hat{A}_{i,t}$ in place of the standard GRPO advantage for every action, jointly optimizing sensing and response from interaction outcomes without a separate sensing reward.

4 Experiments

We first evaluate Self-Focus on proactive and retrospective tasks in Sec. 4.2, and then ablate joint learning and sensing actions in Sec. 4.3. We further probe the quality of its query-agnostic working context on offline Long VideoQA (Sec. 4.4) followed by a runtime analysis. Our closest interaction-model comparison is JoyAI-VL-Interaction (Yao et al., 2026), which is post-trained from the same base model. We reproduce its results where published numbers are unavailable.

4.1 Implementation Details

We initialize Self-Focus from Qwen3-VL-8B-Instruct (Bai et al., 2025) and process every stream at 1fps as default. Each frame is previewed with $N_L = 64$ visual tokens, and `retrieve()` recovers past frames at the same resolution, while `focus()` acquires $N_H = 256$. A single checkpoint serves low, medium, and high context budgets of 2,048, 4,096, and 8,192 visual tokens, specified in the system prompt, which hold at most 32, 64, and 128 frames at preview resolution. We freeze the vision encoder and train the connector and language model on 8×H100 GPUs, first with SFT (AdamW, peak learning rate 10^{-5}, maximum sequence length 16,384), then with RL using $G = 8$ rollouts per group and delayed sensing credit (Sec. 3.5). At inference, each video runs as a single causal session with a rolling KV cache (Xu et al., 2025a; Pang et al., 2026). The system prompt serves as the attention sink, the cache holds only the frames Self-Focus chooses to `keep()` or `focus()`, and frames evicted from the working context leave the cache while the remaining positions stay contiguous. Each query permits a single `retrieve()`. Full details are in Secs. B and D.

4.2 Main results: Streaming Interactions

Benchmarks. For proactive interaction, we evaluate on SPOT-Bench (Chatterjee et al., 2026), ProactiveVideoQA (Wang et al., 2025b), and StreamingBench-PO (Lin et al., 2024). SPOT-Bench is

Table 1: **Proactive Online VideoQA.** We report Timeliness-F1 (%) on SPOT-Bench, PAUC (%) on ProactiveVideoQA, and Top-1 accuracy (%) on StreamingBench. We follow the benchmark-default input rates: 2fps for SPOT-Bench and 1fps otherwise. *reproduced by us

Method	SPOT-Bench		ProactiveVideoQA				StreamingBench
	ABD	PNR	[WEB]	[EGO]	[TV]	[VAD]	PO
Streaming Models							
StreamingVLM-7B (Xu et al., 2025a)	3.9	2.5	-	-	-	-	-
Ctrl-SPOT-8B (Chatterjee et al., 2026)	<u>22.6</u>	<u>19.5</u>	-	-	-	-	-
Interaction Models							
MMDuet2-7B (Wang et al., 2025a)	14.2	5.1	**53.3**	33.6	<u>43.4</u>	<u>28.9</u>	<u>34.7</u>
JoyAI-VL-Interaction-8B (Yao et al., 2026)	17.0	15.6	41.7*	<u>39.2</u>*	43.3*	27.3*	33.6*
Self-Focus-8B (low)	**25.2**	**20.4**	<u>49.4</u>	**46.2**	**47.6**	**30.4**	**40.2**

Table 2: **Retrospective Online VideoQA.**.We report accuracy (%). Input denotes the sampling rate. †SimpleStream uses the latest four frames sampled at 1fps as its input. *reproduced by us

Method	Input	OVO-Bench		StreamingBench
		Real-time	Backward	Real-time
Streaming Models				
TimeChat-Online-7B (Yao et al., 2025)	1fps	61.9	41.7	75.4
HERMES-7B (Zhang et al., 2026)	2fps	73.3	49.3	81.3
SimpleStream-8B (Shen et al., 2026)	4†	81.4	54.0	80.6
DSCache-7B (Pang et al., 2026)	1fps	71.7	44.8	**82.3**
StreamTTT-4B (Chen et al., 2026)	2fps	78.1	59.9	81.3
Interaction Models				
JoyAI-VL-Interaction-8B* (Yao et al., 2026)	1fps	76.9	51.0	81.3
Self-Focus-8B (low)	1fps	**82.6**	55.6	<u>82.2</u>
Self-Focus-8B (medium)	1fps	82.2	<u>64.4</u>	81.9
Self-Focus-8B (high)	1fps	<u>82.3</u>	**66.0**	82.0

our primary benchmark as it evaluates interaction timing over the full video; we use its fully released Detection tasks (ABD/PNR). For retrospective interaction, we use StreamingBench Real-Time and OVO-Bench (Niu et al., 2025), whose Real-time and Backward tasks probe present perception and past recall, respectively. We use SPOT-Bench and OVO-Bench for most ablations, jointly covering evidence from the past, present, and future.

Proactive Online VideoQA. Self-Focus achieves the strongest results on both SPOT-Bench tasks, improving over JoyAI-VL-Interaction by 8.2/4.8 Timeliness-F1 points on ABD/PNR and surpassing the task-specific Ctrl-SPOT-8B (Tab. 1). Since our RL objective optimizes Timeliness-F1, we also evaluate under different metrics: Self-Focus improves macro PAUC over MMDuet2 by 3.6 points on ProactiveVideoQA, leading in three of four domains, and achieves the highest StreamingBench-PO accuracy. Performance varies little across context budgets on these proactive benchmarks, so we report the faster, low-budget setting. Self-Focus sets the state of the art on proactive interaction.

Retrospective Online VideoQA. Here, real-time queries test perception of the present, whereas backward queries additionally require access to past evidence. Together, they expose the perception-memory trade-off (Tab. 2). Across context budgets, Self-Focus maintains nearly unchanged real-time accuracy while backward accuracy rises from 55.6% to 66.0%. The low-budget model achieves the highest OVO real-time accuracy (82.6%), while the high-budget model improves on all reported baselines in both OVO measures, advancing the perception-memory frontier. StreamingBench real-time accuracy remains competitive with the highest reported result (82.2% vs. 82.3%).

4.3 Ablations

Do streaming interaction models need active perception? Yes, when sensing is learned for the interactions it supports (Tab. 3). SFT establishes the streaming policy but performs poorly across all three settings. RL with the interaction reward improves proactive (PNR) timing, yet without learned sensing it falls below the base model on real-time (RT) and backward (BWD) queries, as does JoyAI-VL-Interaction. Adding sensing actions under GRPO (Eq. (1)) restores RT performance, but BWD accuracy remains low because all actions in a rollout share the same trajectory-level advantage. Delayed sensing credit closes this gap, resulting in the only variant that improves over the base model on proactive, real-time, and backward queries simultaneously. As a control, we also train a variant

Table 3: **Learning sensing within interaction policy.** Self-Focus-8B at the low budget, trained on the same data. Gray rows are for reference. † SimpleStream-8B baseline.

Training	Sens.	Inter.	SPOT PNR	OVO RT	OVO BWD
Qwen3-VL-8B†	✗	✗	7.2	81.4	54.0
JoyAI-VL-Interaction-8B	✗	✓	15.6	76.9	51.0
SFT	✓	✓	6.6	71.8	42.5
RL, no sensing (1fps input)	✗	✓	16.9	77.0	49.8
RL, GRPO (Eq. (1))	✓	✓	18.8	82.2	50.5
RL, Eq. (1) + *sensing reward*	✓	✓	16.2	66.2	49.2
RL, Eq. (1) w/ delayed sensing credit	✓	✓	**20.4**	**82.6**	**55.6**

Table 4: **Ablating sensing actions.** Self-Focus-8B at low budget. First row: no sensing actions, every frame at 1fps enters the working context as a preview and the oldest are evicted, similar to other streaming baselines.

`keep()`	`focus()`	`retrieve()`	SPOT PNR	OVO RT	OVO BWD
✗	✗	✗	16.8	76.2	49.4
✓	✗	✗	18.8	77.4	44.3
✓	✓	✗	20.3	81.9	44.2
✓	✗	✓	18.9	78.1	55.2
✓	✓	✓	**20.4**	**82.6**	**55.6**

Table 5: **Offline Long VideoQA** evaluated with Top-1 acc. (%). VideoMME and LongVideoBench are evaluated w/o subtitles. #Frames: the input sampling strategy → the maximum number of frames kept in context when answering a query. *reproduced by us.

Method	#Frames (Input → Kept)	VideoMME Long	VideoMME Overall	EgoSchema Test	MLVU Dev	LongVideoBench Val
Base model, uniform sampling						
Qwen3-VL-8B* (Bai et al., 2025)	1fps → 32	57.1	65.0	68.3	65.5	60.4
Qwen3-VL-8B* (Bai et al., 2025)	1fps → 64	57.3	66.6	68.9	69.6	62.2
Qwen3-VL-8B* (Bai et al., 2025)	1fps → 128	59.9	68.9	69.8	72.6	64.0
Offline keyframe selection						
ReFoCUS-8B (Lee et al., 2026)	– → 32	58.9	68.5	–	–	63.3
FOCUS-7B (Zhu et al., 2026)	– → 64	56.1	65.4	–	–	63.5
BIMBA-7B (Islam et al., 2025)	uniform → 128	–	64.7	71.1	71.4	59.5
Streaming models						
ReKV-7B (Di et al., 2025)	0.5fps → 64	–	–	60.7	68.5	–
Timechat-Online-7B (Yao et al., 2025)	1fps → 1016	52.4	63.3	–	65.4	57.7
Streamo-7B (Xia et al., 2026)	1fps → –	-	67.9	–	–	59.2
HERMES-7B (Zhang et al., 2026)	1fps → –	53.9	64.5	–	60.3	–
Interaction models						
JoyAI-VL-Interaction-8B* (Yao et al., 2026)	1fps → 128†	57.7	66.7	68.1	67.6	56.8
Self-Focus-8B (low)	1fps → 32	59.7	67.8	72.0	70.0	63.0
Self-Focus-8B (medium)	1fps → 64	59.8	68.7	72.2	72.5	64.2
Self-Focus-8B (high)	1fps → 128	**61.3**	**70.9**	**74.8**	**74.5**	**65.2**

with a *sensing reward* that directly rewards each `keep()` or `focus()` within a query's annotated evidence. Unlike delayed sensing credit, this reward encourages acquiring evidence without considering whether it helps answer subsequent queries. The policy exploits it by retaining nearly every frame, sharply degrading real-time and backward accuracy. Overall, these ablations show that *(i)* active perception is essential for streaming interaction, and *(ii)* sensing should be optimized for its downstream interaction outcomes, rather than rewarded in isolation.

What does each sensing action contribute? The actions serve complementary temporal roles (Tab. 4). Enabling `keep()` lets the model selectively manage its bounded working context, improving proactive and real-time performance but weakening backward accuracy resulting in the perception-memory tradeoff phenomenon. Adding `focus()` primarily improves evidence about the present, which further raises proactive and real-time performance, whereas `retrieve()` restores access to past evidence and sharply improves backward accuracy. Using all three yields the strongest result across all settings, allowing a bounded working context to support the present while recovering relevant evidence from the past.

4.4 Probing the Self-Generated Working Context

The most difficult sensing decisions are made before any query is known. During such periods, the policy must decide what keyframes to retain without knowing which future queries they may support. This provides a direct stress test of whether Self-Focus learns to construct a useful query-agnostic working context through delayed sensing credit. We probe this self-generated context on offline long-video benchmarks (Fu et al., 2025; Mangalam et al., 2023; Zhou et al., 2025; Wu et al., 2024), whose videos span several minutes to hours. For each video, Self-Focus performs a single query-agnostic scan at 1fps, choosing to `keep()`, `focus()` or `drop()` each frame, thus generating the working context for the video. Given this context, we test its utility in two ways:

Table 6: **Shared-context probe.** We report accuracy (%), vs. number of search iterations. Self-Focus-8B (medium) is used to match 64 frame budget of the agents. All baseline numbers reproduced by us.

Model	Initialization	VideoMME-Long		LongVideoBench	
		Acc. (↑)	Iters. (↓)	Acc. (↑)	Iters. (↓)
VideoZoomer-7B	Native	52.2	2.19	57.0	2.10
	Self-Focus	54.7	2.16	57.7	2.09
OmniAgent-7B	Native	57.7	9.25	57.5	9.21
	Self-Focus	61.8	5.77	59.4	6.93
GPT-5	Native	75.2	4.25	74.1	3.45
	Self-Focus	**78.0**	2.79	**76.7**	1.95

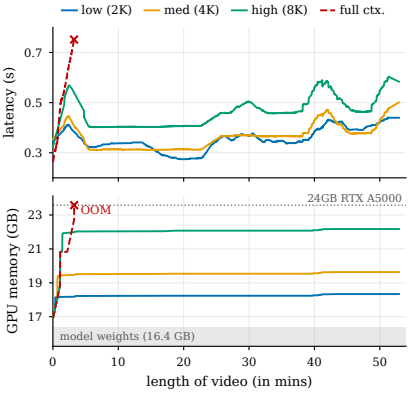


Figure 4: **Self-Focus runtime profile.**

Retrieving from the working context. Here, Self-Focus answers each query using its own `retrieve()` action (Tab. 5). First, as a sanity check, we compare with the base model with uniform sampling at every matched context budget and outperform it across all benchmarks at every budget, confirming that learned sensing allocates the same budget more effectively. More importantly, despite generating the working context causally, Self-Focus at medium and high budgets outperforms recent offline keyframe-selection models with privileged information *i.e.* access to the full video. Finally, among streaming models, Self-Focus achieves the strongest overall performance: even its low-budget variant is competitive with or better than prior methods while retaining only 32 frames at max, and the high variant is the best streaming model across the reported benchmarks.

Reasoning from the shared working context. Here, each frozen video agent starts from the working context of Self-Focus (shared as retained frames) instead of its native initialization, while the agent and its search procedure remain unchanged (Tab. 6). We use VideoMME-Long (VME-L) and LongVideoBench (LVB) because they contain a higher frequency of long videos. The three agents expose complementary search regimes. VideoZoomer's policy effectively collapses to a single search call, so better initialization yields modest accuracy gains at essentially unchanged iterations. OmniAgent searches much more aggressively; initializing it with Self-Focus improves accuracy while reducing search iterations by 38% on VME-L and 25% on LVB. We also reproduce VideoZoomer's search protocol with GPT-5, a stronger perception and reasoning backend. GPT-5 already searches more efficiently, yet our context further improves accuracy while reducing its searches by 34% and 43%. In practical terms, this lowers GPT-5 search cost from \$0.349 to \$0.193 per hour on VME-L and from \$0.704 to \$0.310 on LVB. Overall, Self-Focus complements downstream video agents by providing a query-agnostic working context that accelerates convergence.

4.5 Runtime Profile

We evaluate the runtime profile of Self-Focus for continuous video streaming in Fig. 4. On a single 24GB RTX A5000 desktop GPU, all three variants maintain subsecond per-frame latency throughout the video, with GPU memory bounded by their respective working-context budgets. All variants sustain real-time processing at the standard 1fps (< 1 s latency) input rate, while the low and medium budgets approach 2fps (< 0.5 s latency).

5 Conclusion

A streaming model with bounded context can only respond as well as the evidence available to it at that moment. Self-Focus makes acquiring and preserving this evidence intrinsic to the model's interaction policy. Our ablations show that learning when to respond without learning what to sense improves proactive interaction at the expense of past recall, whereas crediting earlier sensing through future interaction outcomes improves both. The resulting query-agnostic context also benefits frozen long-video agents, showing that evidence acquired before a query is known can support later reasoning. We envision streaming models that continually build and manage their own evidence base across modalities and timescales, supporting interactions whose information needs are not yet known.

Limitations. Self-Focus currently operates on visual frames; extending active perception to audio and other modalities remains future work. Retrieval relies on a CPU-offloaded archive of low-resolution frame KVs, whose retention is managed separately from the learned working context. Future work can extend the learned policy to manage retention across both storage levels.

AI USE STATEMENT

In this work, we used generative AI tools to obtain critical feedback on research methodology and experimental design, and to assist with proofreading and text formatting. We did not use generative AI tools to make research decisions or generate experimental results. We reviewed all AI-assisted work, independently examined the relevant literature, verified AI-assisted formatting changes, and checked the manuscript against our implementation and experimental results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main text and appendix provide the details needed to reproduce our findings. Section 3 describes the model architecture, training objectives, and reward formulation, while Section 4.1 outlines the implementation details. Further implementation details are provided in the appendix. We will release the model checkpoint, training code, and inference code upon acceptance.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International journal of computer vision*, 1(4):333–356, 1988.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pp. 5803–5812, 2017.

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*, 2025.

Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

Leonard Bärmann and Alex Waibel. Where did i leave my keys?—episodic-memory-based question answering on egocentric videos. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1559–1567. IEEE, 2022.

Xinye Cao, Hongcan Guo, Jiawen Qian, Guoshun Nan, Chao Wang, Yuqi Pan, Tianhao Hou, Xiaojuan Wang, and Yutong Gao. Videominer: Iteratively grounding key frames of hour-long videos via tree-based group relative policy optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23773–23783, October 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.

Dibyadip Chatterjee, Zhanzhong Pang, Fadime Sener, Yale Song, and Angela Yao. Don't pause! every prediction matters in a streaming video. *arXiv preprint arXiv:2604.24317*, 2026.

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

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. Livecc: Learning video llm with streaming speech transcription at scale. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29083–29095, 2025.

Joya Chen, Zeyun Zhong, and Mike Zheng Shou. Streamttt: Reconciling real-time perception and long-term memory in streaming vlms. *arXiv preprint arXiv:2608.13416*, 2026.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. *arXiv preprint arXiv:2503.00540*, 2025.

Yang Ding, Xin Lai, Yizhen Zhang, Wei Li, Ruihang Chu, and Yujiu Yang. Videozoomer: Reinforcement-learned temporal focusing for long video reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 20087–20111, 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 2025.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5277–5285. IEEE, 2017.

Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. Frameexit: Conditional early exiting for efficient video recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 15608–15618. Computer Vision Foundation / IEEE, 2021.

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.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 3302–3310, 2025.

Md Mohaiminul Islam, Tushar Nagarajan, Huiyu Wang, Gedas Bertasius, and Lorenzo Torresani. Bimba: Selective-scan compression for long-range video question answering. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29096–29107. IEEE, 2025.

Hosu Lee, Junho Kim, Hyunjun Kim, and Yong Man Ro. Refocus: Reinforcement-guided frame optimization for contextual understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8291–8302, 2026.

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

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

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

Zikang Liu, Longteng Guo, Handong Li, Ru Zhen, Xingjian He, Ruyi Ji, Xiaoming Ren, Yanhao Zhang, Haonan Lu, and Jing Liu. Thinking in streaming video. In *European Conference on Computer Vision*, pp. 21–38. Springer, 2026.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Hongzi Mao, Shaileshh Bojja Venkatakrishnan, Malte Schwarzkopf, and Mohammad Alizadeh. Variance reduction for reinforcement learning in input-driven environments. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hyg1G2AqtQ.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18902–18913, 2025.

Zhanzhong Pang, Dibyadip Chatterjee, Fadime Sener, and Angela Yao. Decouple and cache: Kv cache construction for streaming video understanding. *arXiv preprint arXiv:2605.01858*, 2026.

Yaxuan Qin, Hefei Li, Wenqi Mu, and Yancheng He. Efficient frame selection for long video understanding via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16944–16953, June 2026.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2613–2622. IEEE, 2020.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128. IEEE, 2025.

Thinking Machines. Interaction Models: A Scalable Approach to Human-AI Collaboration, 5 2026. URL https://thinkingmachines.ai/blog/interaction-models/.

Shaoguang Wang, Weiyu Guo, Ziyang Chen, Yijie Xu, Xuming Hu, and Hui Xiong. Less is more: Token-efficient video-qa via adaptive frame-pruning and semantic graph integration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9856–9866, 2026a.

Yueqian Wang, Songxiang Liu, Disong Wang, Nuo Xu, Guanglu Wan, Huishuai Zhang, and Dongyan Zhao. Mmduet2: Enhancing proactive interaction of video mllms with multi-turn reinforcement learning. *arXiv preprint arXiv:2512.06810*, 2025a.

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*, 2025b.

Ziyang Wang, Honglu Zhou, Shijie Wang, Junnan Li, Caiming Xiong, Silvio Savarese, Mohit Bansal, Michael S Ryoo, and Juan Carlos Niebles. Active video perception: Iterative evidence seeking for agentic long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9088–9099, 2026b.

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.

Wenhao Wu, Dongliang He, Xiao Tan, Shifeng Chen, and Shilei Wen. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6221–6230. IEEE, 2019a.

Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1278–1287. IEEE, 2019b.

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

Zhenghao Xing, Ruiyang Xu, Yuxuan Wang, Jinzheng He, Ziyang Ma, Qize Yang, Yunfei Chu, Jin Xu, Junyang Lin, Chi-Wing Fu, et al. Native active perception as reasoning for omni-modal understanding. *arXiv preprint arXiv:2606.19341*, 2026.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*, 2025a.

Ziqiang Xu, Qi Dai, Tian Xie, Yifan Yang, Kai Qiu, DongDong Chen, Zuxuan Wu, and Chong Luo. Viarl: Adaptive temporal grounding via visual iterated amplification reinforcement learning. *arXiv preprint arXiv:2505.15447*, 2025b.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. Vca: Video curious agent for long video understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20168–20179. IEEE, 2025.

Dingyu Yao, Junhao Zhou, Chenxu Yang, Chuanyu Qin, Haowen Hou, Zheming Liang, Congcong Wang, Yuhang Cao, Shenglong Ye, Shuai Xie, et al. Joyai-vl-interaction: Real-time vision-language interaction intelligence. *arXiv preprint arXiv:2606.14777*, 2026.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. *arXiv preprint arXiv:2504.17343*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

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

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22056–22065, 2025.

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

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

Zirui Zhu, Hailun Xu, Yang Luo, LIU YONG, Kanchan Sarkar, Zhenheng Yang, and Yang You. Focus: Efficient keyframe selection for long video understanding. In *International Conference on Learning Representations*, volume 2026, pp. 98463–98486, 2026.

Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. Videolucy: Deep memory backtracking for long video understanding. *Advances in Neural Information Processing Systems*, 38:25660–25691, 2026.

Appendix

Table of Contents

A Glossary 15

B Additional Implementation Details 15

C Additional Benchmark Details 16

D Additional Training Details 17

D.1 Streaming Episodes 17

D.2 Supervised Fine-Tuning 17

D.3 Reinforcement Learning 17

D.4 Analysis of Delayed Sensing Credit 18

E System Prompt 19

E Qualitative Examples 19

A GLOSSARY

We summarize the key concepts used throughout the paper.

TERM	DEFINITION
Sensing	Acquiring and retaining visual evidence from the stream.
Interaction Model	A streaming model that decides when to respond or remain silent.
Active Perception	Actively controlling what evidence to acquire and retain.
Native Active Perception	Sensing actions are learned by the model.
Dead-time	Periods with no active query, requiring query-agnostic sensing.
Working Context	Bounded evidence available to the policy at a streaming decision.
Shared Context	Acquired evidence exported to a downstream agent.
Streaming Rollout	A causal trajectory executing the policy's own sensing and response actions.
Exogenous Stream	A video stream that progresses independently of the model's actions.
Shared Prefix	A common initial state reconstructed by replaying reference actions before RL branching.
Delayed Sensing Credit	An estimate of an action's advantage in subsequent interaction return.

B ADDITIONAL IMPLEMENTATION DETAILS

Context budgets and reference sensing. Tab. 7 lists the three context budgets and the budget-conditioned rules that generate reference sensing actions (Sec. D.1). We first establish a 24 GB GPU memory constraint and choose the context budgets to support 32/64/128 preview frames, matching the evaluation settings of existing video models for fair comparison. The frame resolutions, with nominal token counts N_L and N_H, are chosen accordingly rather than tuned specifically for Self-Focus. A `focus()` costs four preview frames, and a retrieved frame costs one preview frame. The budget bounds only the visual tokens in the working context; text tokens (timestamps, queries, and responses) do not count against it.

Table 7: Working context budgets and the maximum number of frames they hold at preview (N_L) and high (N_H) resolution.

Budget	Low	Medium	High
B (tokens)	2,048	4,096	8,192
Max. frames at N_L	32	64	128
Max. frames at N_H	8	16	32

Efficient Streaming Inference with Rolling KV cache. Following StreamingVLM (Xu et al., 2025a) and SPOT-Bench (Chatterjee et al., 2026), we keep one KV cache per stream and never recompute it from scratch. The system prompt Sec. E forms the attention sink and is never evicted. After each preview, `drop()` removes it from the cache, `keep()` commits it unchanged, and `focus()` replaces it with its N_H encoding. `retrieve()` reads low-resolution frame KVs from a CPU-offloaded archive (similar to ReKV (Di et al., 2025)) and appends them to the cache together with queries and answered turns. A query stays active for 120 s, following (Chatterjee et al., 2026), and its retrieved frames are removed once it expires. When the working context exceeds B, the oldest retained frames are evicted with their KV entries, and the remaining tokens are re-indexed to contiguous M-RoPE positions. Sensing action calls are not committed into the KV: after each sensing action, the cache is rolled back to trim the action.

C Additional Benchmark Details

C.1 Online VideoQA - ProActive and Retrospective

SPOT-Bench (Chatterjee et al., 2026) tests whether a streaming model responds at the right moment as events unfold, and correclty stays silent the rest of the time. We use its two detection tasks; Action Boundary Detection (ABD) asks the model to flag when an action starts or ends (602 videos). Point-of-No-Return Detection (PNR) asks it to flag the moment an object's state changes irreversibly (297 videos). Videos average about 4 minutes and contain several queries each, some needing more than one response. We report it's Timeliness-F1 metric. Each ground-truth response has a short time window set from a human study (±1 s for ABD, ±0.5 s for PNR). Correct responses inside the window get full credit, early or late ones get partial credit, and missed or extra responses lower the F1 score.

ProActiveVideoQA (Wang et al., 2025b)A question is given at the start of the video, and the model must decide on its own when to answer as relevant evidence appears, sometimes more than once. The benchmark has 1,427 open-ended questions over 1,377 videos in four domains: web videos [WEB] (17 s), egocentric cooking/tasks [EGO] (6 min), TV series with dialogue [TV] (76 s), and surveillance anomaly detection [VAD] (2 min). We report the PAUC metric. An LLM judge scores the model's answers over time inside each ground-truth window, and PAUC is the area under that score curve. Answers that are both early and correct score higher.

StreamingBench (Lin et al., 2024). We evaluate our approach on two different splits of the benchmark, each probing a specific capability. In **Proactive Output (PO)**, the model is given a condition in advance and must produce the output at the right time proactively. The evaluated set has 50 questions. A response counts as correct only if it comes within 2 seconds of the ground-truth time. **Real-Time (RT)** tests perception of what is happening at the moment the query is posed. It has 2,500 multiple-choice questions with four options each on 500 videos and the model sees the video only up to the question time. We use accuracy as the evaluation metric for both the splits.

OVO-Bench (Niu et al., 2025) asks questions at specific points during a video and checks whether the model can use the right part of the stream to answer. It has 644 videos (ranging from a few minutes up to about 30 minutes) and around 2,800 questions. We use two of its three modes. **Real-Time Visual Perception** has 6 tasks about the current moment: object, attribute, action, and text recognition, spatial understanding, and future prediction. **Backward Tracing** has 3 tasks that require recalling the past: episodic memory, action sequence, and hallucination detection. Both are multiple choice.

C.2 Long VideoQA

EgoSchema (Mangalam et al., 2023) probes understanding of long egocentric videos, where the answer depends on watching a large portion of the clip rather than a single frame. It has about 5,000 five-option multiple-choice questions, each on a 3-minute egocentric clip from Ego4D. On average, the model is required to watch roughly 100 seconds of video to provide an answer. Accuracy is the evaluation metric we report.

VideoMME (Fu et al., 2025) is a general video understanding benchmark covering perception and reasoning across 6 domains and 30 categories. It has 900 videos and 2,700 four-option multiple-choice questions, three per video. The videos are split into short (under 2 min), medium (4–15 min), and long (30–60 min) sets. We report accuracy on the long and overall set, without subtitles.

MLVU (Zhou et al., 2025) tests long-video understanding at three levels: the whole video (e.g., topic, anomaly), a single detail (e.g., needle QA, plot QA), and multiple details (e.g., action order, action count). It has 1,730 videos ranging from 3 minutes to over 2 hours (about 15 minutes on average). We report multiple-choice accuracy on the dev set, which has four options per question.

LongVideoBench (Wu et al., 2024) consists of queries targeting the model's capability to accurately retrieve and reason over long videos. Each question first describes a particular scene and then queries about it. To answer, the model has to search for the particular scene in the video and reason about it. The benchmark has 6,678 multiple-choice questions on 3,763 videos up to one hour long. We report accuracy on the validation set (1,337 questions, 752 videos), without subtitles.

D Additional Training Details

D.1 Streaming Episodes

Data Source. We source temporal grounding annotations from Charades-STA Gao et al. (2017), QAEgo4D Bärmann & Waibel (2022), QVHighlights Lei et al. (2021), and DiDeMo Anne Hendricks et al. (2017), alongside point-wise annotations from FineGym Shao et al. (2020). We supplement these with QA annotations from VTG-LLM Guo et al. (2025) and Streamo Xia et al. (2026), and incorporate offline VideoQA data from LLaVA-Video Zhang et al. (2024), following existing streaming models.

Queries. Each annotated query with evidence interval $[s, e)$ is placed in the stream at an arrival time τ. Short evidence ($e - s \leq 5$s) yields a proactive query that arrives 10–60s before s, with response window $[s, e + 3)$. Longer evidence yields a retrospective query that arrives after e, with a delay drawn from immediate (0–3s), near (8–30s), or far (60–600s) regimes in a 30/40/30 ratio, or, for 30% of these examples, a mixed query that arrives 2-5s before e, with response window $[e - 1, e + 3)$. Video-level QA (Zhang et al., 2024) is posed after the last frame of its clip. Half of all queries are four-way multiple choice, with distractors drawn from answers in the same video. Queries from the same video form one multi-turn episode. All offsets are hashed per example, so the construction is deterministic.

Reference actions. Outside the evidence of an active query, the reference keeps each frame independently with probability p = 1/16, 1/8, and 1/4 under the low, medium, and high budgets, and drops the rest. Within an annotated evidence, the reference focuses each frame, replacing a share ρ with `keep()`, and keeps a 2s band on either side of proactive evidence. Retrospective evidence is observed before its query is known, so the reference focuses only a few keyframes within it, sampled from a Gaussian centred on the evidence with $\sigma = \max(1, (e - s)/4)$; these are sampled to anchor retrieval. When a query arrives, the reference calls the null `retrieve()` if the working context already holds the evidence, and otherwise retrieves the interval that maximizes its overlap with the missing evidence minus a token cost.

D.2 Supervised Fine-Tuning

Sampling. Each sampled segment yields up to 12 single-step instances. A quarter is drawn at the natural action frequency; the rest are drawn in order from response (25%), retrieval call (10%), null retrieval (8%), proactive wait-response chains (6%), pending silence between a query's arrival and its response window (5%), and sensing, split 40/30/30 over `drop()`, `keep()`, and `focus()`, each rate applying to draws not taken by the previous pools. The context budget is drawn per segment and sets both the replayed working context and the targets. Earlier queries in the episode appear as dialogue history, hidden in 25% of segments.

On-policy states. Reference replay only shows the model working contexts built by the reference actions, whereas at inference it acts on its own. Following DAgger (Ross et al., 2011), we roll out the SFT policy from each query's arrival on training videos and label every visited state with the correct action: a premature response is relabelled `<silent>` and the rollout continues from the corrected state, a state whose working context holds at least half of the evidence becomes a response target, and states after a response become `<silent>` targets. These states form half of each SFT epoch, mixed 50/40/10 over premature, pending, and ready states.

Objective. Only the model's actions and answers are supervised; the system prompt, observations, queries, dialogue history, and retrieved frames are masked. The language-modeling loss is averaged separately over sensing, response-action, and answer tokens and summed. At the token that decides each action, we add a cross-entropy weighted by the inverse frequency of the target class among sampled instances, normalized to mean one and capped at 8.

D.3 Reinforcement Learning

Segments and prefixes. Each group samples a segment $[t_0, t_1)$ of an episode with $K \in \{1, 2, 3\}$ queries whose response windows do not overlap. It starts 2-10s before the earliest evidence and ends 6s after the last response window, so all evidence lies inside the segment, where sampled actions can be credited for it; evidence before t_0 would sit in the shared prefix. Segments longer than $L_{\max} = 256$s are discarded, and at least 30% of segments contain a query whose evidence

ends more than B/N_L seconds before it arrives. The prefix before t_0 replays the reference actions under the segment's budget, so the working context is typically full and new commitments evict older ones. To expose recovery states, 25% of prefixes drop half of the reference `keep()` actions, 15% drop all of them, and 30% end with 30s sensed by the policy itself.

Reward terms. R_{task} is option-letter accuracy for multiple-choice queries and $\max(\text{EM},\ \text{token-F1})$ otherwise. For $R_{\text{T-F1}} = 2\text{TP}/(2\text{TP}+\text{FP}+\text{FN})$, a response at step t is a true positive if it falls in an unmatched window $[a, b)$, *i.e.* $a \le t < b$, and its content score is at least 0.5. Unlike PAUC (Wang et al., 2025b), which scores response content with an LLM judge, T-F1 is computed from annotations alone and penalizes both over- and under-responding. We score each query separately: its task, T-F1, and format terms are computed over the responses made while it is active, and the rollout's reward is their sum, so a query's score is fixed once its response window closes. Responses made while no query is active count as false positives in a separate term, fixed at the end of the rollout. R_{fmt} is the fraction of valid emissions, minus penalties for grammar violations and for collapse onto a single action. $C_{\text{ctx}} = (N_{\text{keep}} + N_{\text{focus}} + N_{\text{retrieve}})/B$ counts committed visual tokens; the preview is identical across rollouts and is not charged. We also subtract 0.3 when the first response precedes the response window and 0.3 when a window passes without a response; without the latter, the policy collapses to permanent silence. We use $\lambda_{\text{task}} = 1$, $\lambda_{\text{time}} = 0.5$, $\lambda_{\text{fmt}} = 0.2$, and $\lambda_{\text{ctx}} = 0.1$.

Optimization. We use $G = 8$ rollouts at temperature 1.0 and top-p 1.0, with per-rollout seeds. We skip groups in which all rollouts obtain the same return after every step, since all their credits are zero (Yu et al., 2026); equal total returns alone do not suffice, as two queries scored $(1, 0)$ and $(0, 1)$ show. Credits are not divided by the group standard deviation. We score the exact sampled tokens and perform one update per batch, so the policy ratio stays at one and clipping is inactive. We use no KL penalty and an entropy bonus of 10^{-3}, normalize each emission's log-likelihood by a fixed 64 tokens, and optimize with Adafactor at learning rate 10^{-6} and gradient clipping at 1.0. Groups are drawn over query type, the delay between evidence and query, and budget, with a curriculum that admits longer delays every 12 iterations.

D.4 Analysis of Delayed Sensing Credit

We show that delayed sensing credit (Eq. (2)) yields an unbiased policy gradient, even though the working contexts of the rollouts diverge after the shared prefix.

Setup. Let $c = (p, z, B)$ collect the shared prefix p, the recorded frames and query schedule z (including the annotations used for scoring), and the context budget B. At each frame and each query arrival, the policy emits a sensing action (`retrieve()`, possibly null) followed by a response action; we index these actions in order by $t = 1, \dots, T$. Their number and order are fixed by c, so step t refers to the same point of the stream in every rollout. Each a_t^i is sampled from $\pi_\theta(\cdot \mid h_t^i)$, where h_t^i is the model's input at that step (current observation, working context, dialogue history, and budget), determined by c and the earlier actions of o_i. Given c, the $G \ge 2$ rollouts are sampled independently, and $\psi_t^i = \nabla_\theta \log \pi_\theta(a_t^i \mid h_t^i)$ satisfies $\mathbb{E}[\psi_t^i \mid h_t^i, c] = 0$. Let R^i be the return of o_i and $D_{<t}^i$ its contributions fixed before step t, *i.e.* the scores of queries whose response windows closed before t and the cost of tokens committed before t. Then $R_{>t}^i = R^i - D_{<t}^i$ includes the cost of the action at t, and $\hat{A}_{i,t} = R_{>t}^i - b_{i,t}$ with $b_{i,t} = \frac{1}{G-1}\sum_{j \ne i} R_{>t}^j$.

Proposition 1. *Assume that $P(c)$ and the reward do not depend on θ, and the standard regularity conditions of the score-function identity. Then $b_{i,t}$ is an unbiased estimate of $\mu_t(c) = \mathbb{E}[R_{>t} \mid c]$ with variance $\text{Var}(R_{>t} \mid c)/(G-1)$, and, treating sampled returns and references as constants during differentiation,*

$$\mathbb{E}\Big[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{T}\psi_t^i \hat{A}_{i,t}\Big] = \nabla_\theta\, \mathbb{E}[R]. \tag{3}$$

Proof. Given c, the other rollouts give $G-1$ independent samples of $R_{>t}$, which yields the mean and variance of $b_{i,t}$. They are independent of o_i, so $\mathbb{E}[\psi_t^i b_{i,t} \mid c] = \mathbb{E}[\psi_t^i \mid c]\,\mathbb{E}[b_{i,t} \mid c] = 0$. Let $\mathcal{F}_{t-1}^i$ contain c and the actions of o_i before t. Since $D_{<t}^i$ is determined by $\mathcal{F}_{t-1}^i$ and a_t^i is sampled causally, $\mathbb{E}[\psi_t^i D_{<t}^i] = \mathbb{E}\big[D_{<t}^i\, \mathbb{E}[\psi_t^i \mid \mathcal{F}_{t-1}^i]\big] = 0$. Hence $\mathbb{E}[\psi_t^i \hat{A}_{i,t}] = \mathbb{E}[\psi_t^i R^i]$, and summing over t and averaging over i gives Eq. (3) by the score-function identity with $P(c)$ fixed. □

Not a same-state advantage. Let $Q_t^c(h,a) = \mathbb{E}[R_{>t} \mid h_t = h, a_t = a, c]$, $V_t^c(h) = \mathbb{E}[R_{>t} \mid h_t = h, c]$, and $A_t^c = Q_t^c - V_t^c$. Since $b_{i,t}$ is independent of o_i given c,

$$\mathbb{E}\big[\hat{A}_{i,t} \mid h_t^i, a_t^i, c\big] = A_t^c(h_t^i, a_t^i) + \big(V_t^c(h_t^i) - \mu_t(c)\big). \tag{4}$$

The second term reflects the differences in input, such as working context and dialogue history, that earlier actions of o_i produced. It vanishes at the first step after the shared prefix, where h_t is determined by c, and need not vanish once rollouts diverge. It does not depend on a_t^i, so its expected product with ψ_t^i is zero and the gradient stays unbiased, but $\hat{A}_{i,t}$ is not the advantage of a_t^i at a matched state. The analysis conditions on the recorded future; the policy does not observe it.

Credit before a query arrives. Sensing actions taken before a query arrives affect its score through the inputs of later actions, so later query outcomes train earlier sensing while the policy remains causal. Response windows close at the same step in every rollout, so between two steps at which no window closes, the credits within a rollout differ only by the context cost committed in between. The credit thus supplies delayed outcome feedback without attributing it to individual frames.

Scope. Proposition 1 concerns the unnormalized on-policy gradient with all actions included and $P(c)$ fixed; for prefixes that end with a stretch sensed by the policy, it omits the term through which θ changes the prefix. The fixed 64-token normalization only rescales the gradient, and skipped groups remain in the loss denominator, so skipping removes only zero terms. The entropy bonus, gradient clipping, and the optimizer fall outside the stated equality, and we make no claim of minimum variance.

E System Prompt

The system prompt is constant for every episode. No special tokens are added to the vocabulary: the actions, control symbols, and stream markers are plain text, and timestamps use Qwen3-VL's own `<t seconds>` marker.

```
You are Self-Focus, a real-time video assistant observing a continuous camera feed frame
by frame. Each frame is labeled with its absolute time as <t seconds>; the last frame is
the current moment. Frames reach you as low-resolution previews, and your context is
bounded, so you must decide what to remember, what to look at closely, and when to speak.

## Action format
At every step, output exactly one sensing action, then exactly one control action. A step
is either a new frame or the arrival of a query.

Sensing at a frame (what to do with the current frame):
  drop()   Discard the preview. Choose this when the frame adds nothing to the active
  query or to what is worth remembering.
  keep()   Commit the low-resolution preview to context. Choose this when the frame may
  matter later but detail is not needed.
  focus()  Re-observe the current frame at high resolution and commit it. Choose this
  when fine detail is needed right now; the high-resolution view arrives before your
  control action.

Sensing at a query (only when a query arrives; what to look back at, now that you know
the question):
  retrieve(t_s,t_e)  Re-observe seconds t_s to t_e of the past as low-resolution
  previews. Choose this when the query concerns something already seen that your context
  no longer holds. The retrieved frames arrive before your control action.
  retrieve()         Do not retrieve. Choose this when your context already holds what
  the query needs, or when the answer lies ahead in the stream.

Respond (what to do about the active query):
  <silent>            No response is needed now; keep observing.
  <response> <text>   The queried event is happening now, or the evidence in your context
  is sufficient. Answer concisely; if the query lists lettered options, answer with the
  letter.

Retrieval costs time and compute; commit and retrieve only what is likely to be useful,
and answer only when the evidence is sufficient.

Context budget: medium.
```

F Qualitative Examples

We visualize how Self-Focus allocates `focus()` action under query-agnostic and query-conditioned settings in Fig. 5, and illustrate the importance of `focus()` for understanding the present in Fig. 6.

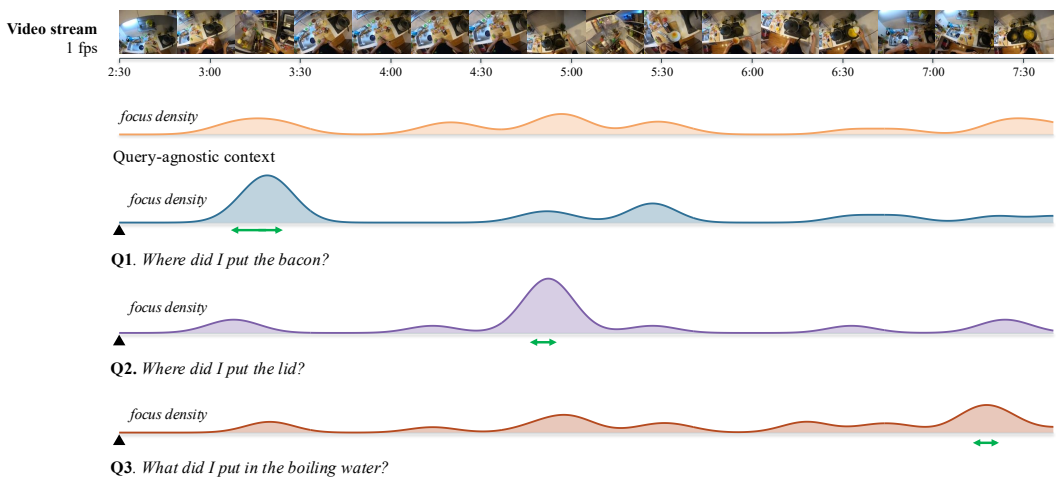


Figure 5: **Query-agnostic versus query-conditioned evidence acquisition.** An Ego4D-NLQ example comparing the temporal distribution of `focus()` actions during a query-agnostic scan (top) with separate query-conditioned scans for three proactive queries. Green arrows indicate the annotated evidence. With a known query, sensing concentrates around query-relevant evidence; without one, the policy distributes its sensing budget across potentially informative moments, performing online keyframe selection to maximize expected utility for future interactions.

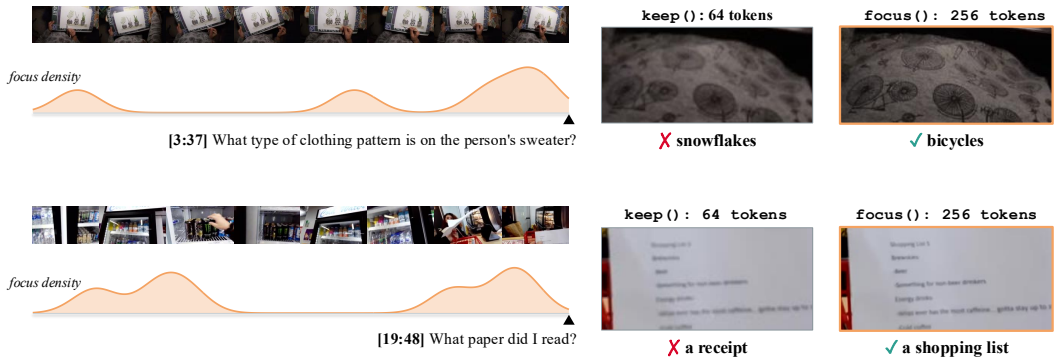


Figure 6: **High-resolution sensing for present perception.** Two examples from OVO-Bench Real-Time Visual Perception. The focus-density curves illustrate the temporal allocation of high-resolution observations. Compared with retaining the low-resolution preview using `keep()` ($N_L = 64$ tokens), `focus()` ($N_H = 256$ tokens) reveals fine-grained details needed to solve difficult perception tasks.

We measure the temporal distribution of `focus()` actions using Gaussian-smoothed focus density,

$$\rho_{\mathcal{F}}(t) = \frac{1}{|\mathcal{F}|} \sum_{\tau \in \mathcal{F}} \mathcal{N}(t; \tau, \sigma^2), \qquad \mathcal{F} = \{\tau : a_\tau = \texttt{focus()}\}, \tag{5}$$

where a_τ denotes the sensing action at time τ. Normalization makes the curves comparable in terms of temporal allocation, independently of the total number of `focus()` actions. We use $\sigma = 8\,\text{s}$ for the query-agnostic comparison and $\sigma = 2.5\,\text{s}$ for the present-perception examples.