RETRIEVE-THEN-REASON: ALWAYS-ON PROACTIVE ASSISTANCE WITH ON-DEMAND REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Always-on proactive AI assistants must continuously handle streaming multimodal information, but invoking a multimodal large language model (MLLM) at every timestep is prohibitively expensive for on-device deployment. To address this challenge, we reformulate continuous proactive monitoring as a lightweight semantic retrieval problem and present PRISM (Proactive Reasoning via Semantic Monitoring), a cascaded framework that separates continuous retrieval from on-demand contextual reasoning. PRISM first identifies potentially relevant events with high recall and then filters them according to their contextual actionability, reducing false alarms and unnecessary MLLM invocations. Specifically, lightweight Visual and State Retrievers continuously retrieve instruction-relevant visual evidence and semantic states from the incoming multimodal stream leaving contextual actionability assessment to the subsequent verification stage. On-demand Verifier then evaluates whether each retrieved candidate satisfies the requested condition using the current multimodal context, recent history, and relevant semantic states before forwarding it to the MLLM-based Responder. On the PhoStream benchmark, PRISM improves performance on future-dependent proactive tasks, substantially reduces premature responses, and achieves a performance–energy Pareto improvement over direct MLLM invocation at every step. Our results show that lightweight semantic monitoring combined with contextual verification enables efficient continuous proactive assistance while reserving expensive reasoning for potentially actionable events.

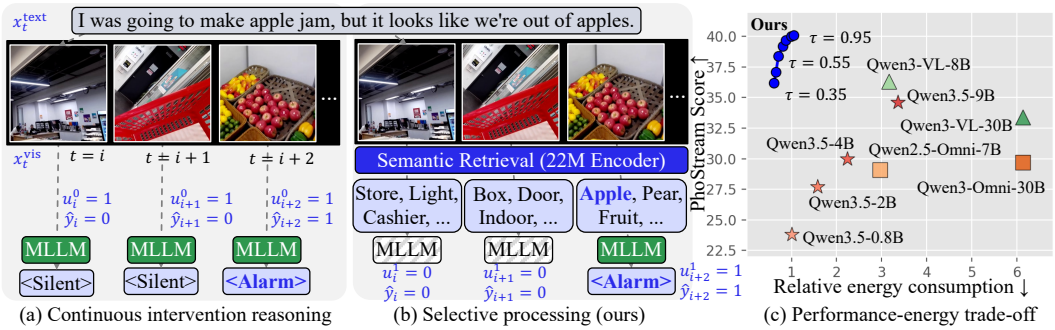
1 INTRODUCTION

Artificial intelligence assistants are moving beyond reactive question answering toward proactive systems that monitor a user's surroundings and provide assistance when it becomes useful (Xie et al., 2026; Pu et al., 2025). Wearable and mobile devices make this interaction increasingly practical by providing continuous visual and conversational observations (Khan & Maes, 2021; Huang et al., 2025). However, proactive assistance introduces a demanding systems problem: before generating any response, the assistant must continuously determine whether the current situation warrants intervention (Xie et al., 2026).

A straightforward solution is to repeatedly process each incoming observation window with a capable multimodal large language model (MLLM) (Team, 2026; Bai et al., 2025), jointly deciding whether to intervene and what to say. As illustrated in Fig. 1(a), this design couples always-on perception with decoder-scale reasoning and is prohibitively expensive for sustained operation on battery-constrained edge devices (see Fig. 6 in Appendix A.1). Recent systems reduce this burden through adaptive visual processing, temporal compression, or cascaded inference (Chen et al., 2025; Yu et al., 2025; Xu et al., 2026). For example, LlamaPIE (Chen et al., 2025) uses a 1B language model to decide whether an 8B response generator should be invoked. This makes response generation selective, but still leaves billion-parameter intervention reasoning as an always-on workload. In long-running streams, from always-on wearable assistants used throughout daily life, useful intervention opportunities may be sparse (Kalnikaite et al., 2010), causing the cost of this continuous gate to accumulate over time.

We argue that only lightweight perception should operate continuously, while language-model reasoning should be invoked selectively. To realize this principle, we cast always-on proactive assistance as a *retrieve–verify–respond* process. Because intervention opportunities are sparse, a lightweight encoder-based *Semantic Retriever* first identifies observations that may contain request-

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.



(a) Continuous intervention reasoning (b) Selective processing (ours) (c) Performance-energy trade-off

Figure 1: (a) Conventional continuous reasoning repeatedly invokes an MLLM over the incoming stream to determine whether intervention is warranted. (b) PRISM continuously retrieves semantically relevant evidence using lightweight encoders and invokes downstream reasoning only for selected candidates. (c) Performance versus estimated inference energy on PhoStream.

relevant evidence. Because retrieval establishes semantic relevance rather than actionability, a sub-1B *Verifier* contextually re-scores the retrieved candidates and determines whether the requested condition is currently satisfied. Finally, because MLLM-based response generation is the most expensive stage in our pipeline, only verified candidates activate the *Responder*. This selective cascade, illustrated in Fig. 1(b), applies inexpensive encoding continuously, lightweight verification selectively, and expensive generation only to rare actionable events. As a result, it achieves a substantially improved performance–energy trade-off compared to always-on MLLM processing, as shown in Fig. 1(c).

Efficient first-stage filtering must support dynamically specified requests without training a task-specific detector for each request. We therefore formulate candidate discovery as semantic retrieval rather than fixed-class detection. For visual monitoring, the active request is converted into an instruction-conditioned interest set and matched against objects and places in each incoming frame. For conversational monitoring, ongoing speech is mapped into three different states: 1) proactive states that suggest possible assistance, 2) pending states that indicate the system should wait, and 3) user states that provide situational context. However, semantic relevance alone does not establish that intervention is currently warranted. For example, under the request "Tell me when I leave my keys behind," observing the keys is relevant but insufficient to trigger an intervention, and doing so would cause premature responses and unnecessary downstream computation. We refer to these non-actionable downstream activations as *false alarms*, distinguishing them from user-facing false interventions. The Verifier suppresses such candidates using the current multimodal context, recent history, pending-state evidence, and its confidence that the requested condition has occurred.

On PhoStream (Lu et al., 2026), PRISM achieves the highest overall score (40.0) while reducing normalized inference cost by over 3× relative to the best-performing baseline (1.00 vs. 3.16). It improves Forward performance from 7.7 to 22.7 and reduces premature responses from 81.3% to 9.4%. While existing benchmarks, including PhoStream, primarily evaluate proactive assistance under explicit user requests, proactive assistants must often identify intervention opportunities that are never directly stated (e.g., prompt in Fig. 1). To assess a more realistic setting in which intervention opportunities must be *inferred* rather than *stated*, we introduce *SIR-Bench*, a complementary benchmark focused on *implicit triggering*, where the assistant must infer latent user needs from visual observations and conversational context before deciding whether to intervene.

Our contributions are summarized as follows:

- We formulate always-on proactive assistance as a retrieve–verify–respond process that makes both intervention reasoning and response generation selective in sparse-event streams.
- We realize lightweight candidate discovery through instruction-conditioned visual retrieval and structured retrieval of proactive, pending, and user states from conversation.
- We develop a sub-1B contextual Verifier that addresses the relevance–actionability gap by re-scoring retrieved evidence using multimodal context and recent history.
- We demonstrate improved intervention timing and estimated energy efficiency on PhoStream, component-level false-alarm reductions, and introduce SIR-Bench to evaluate intervention reasoning from implicit multimodal cues without explicit requests.

2 PROBLEM FORMULATION

2.1 PROACTIVE ASSISTANCE WITH SELECTIVE PROCESSING

Problem setting. Consider a system that continuously observes a multimodal stream $\mathcal{X}_{1:T} = (x_1, \ldots, x_T)$, where $x_t = (x_t^{\text{text}}, x_t^{\text{vis}})$ contains the textual and visual observations available at time t (Fig. 1). Let $y_t \in \{0, 1\}$ denote the ground-truth intervention label, where $y_t = 1$ indicates that the current situation warrants intervention and $y_t = 0$ indicates that the system should remain silent. The system predicts an intervention decision $\hat{y}_t \in \{0, 1\}$. When $\hat{y}_t = 1$, it generates a response $\hat{o}_t$; otherwise, it remains silent. A false-positive intervention occurs when $\hat{y}_t = 1$ and $y_t = 0$, while a false-negative intervention occurs when $\hat{y}_t = 0$ and $y_t = 1$.

Selective processing. To avoid applying expensive reasoning uniformly across the stream, the system processes each observation through a sequence of selective stages: $u_t^{(0)} \to u_t^{(1)} \to \cdots \to u_t^{(L)} \to \hat{y}_t \to \hat{o}_t$. Here, $u_t^{(i)} \in \{0, 1\}$ indicates whether observation t is retained after processing stage i, and L is the number of intermediate stages. Every observation initially enters the pipeline, so $u_t^{(0)} = 1$. An observation is forwarded to stage $i + 1$ only when $u_t^{(i)} = 1$. After the intermediate stages, $\hat{y}_t$ determines whether the system generates a response $\hat{o}_t$.

Processing metrics. For processing stage i, let $\rho_i = \Pr\left(u_t^{(i)} = 1 \mid y_t = 1, u_t^{(i-1)} = 1\right)$ denote its conditional true-positive retention rate, and let $f_i = \Pr\left(u_t^{(i)} = 1 \mid y_t = 0, u_t^{(i-1)} = 1\right)$ denote its conditional false-positive activation rate. Thus, ρ_i measures how often stage i preserves observations for which intervention is warranted, whereas f_i measures how often it forwards observations that do not require intervention. The corresponding conditional false-rejection rate is $\text{FRR}_i = 1 - \rho_i$, which measures the fraction of intervention-requiring observations reaching stage i that are rejected at that stage.

We distinguish false-positive activations from false-positive interventions. A false-positive intervention is a user-facing error ($\hat{y}_t = 1, y_t = 0$), whereas a false-positive activation occurs when an observation that does not require intervention is forwarded to a subsequent processing stage. Although such an observation may ultimately be rejected, forwarding it still incurs additional downstream computation. We refer to these false-positive activations as *false alarms*.

2.2 WHY FALSE-ALARM REDUCTION MATTERS

False-positive computation overhead. Let $p = \Pr(y_t = 1)$ denote the intervention rate. In the always-on setting considered in this work, intervention opportunities are sparse, so $p \ll 1$ (Kalnikaite et al., 2010). Let e_i denote the energy required to process one observation at stage i, and let e_R denote the energy cost of the final reasoning stage. Even if observations that do not require intervention are ultimately rejected, false-positive activations can propagate them through the pipeline and incur unnecessary downstream computation. For a system with L selective processing stages, the expected downstream energy spent on these observations is

$$\mathcal{W}_{\text{FP}} = (1 - p) \sum_{i=1}^{L} \left(\prod_{j=1}^{i} f_j \right) e_{i+1}, \tag{1}$$

where $e_{L+1} = e_R$. The term $\prod_{j=1}^{i} f_j$ is the probability that an observation not requiring intervention passes through the first i selective stages and incurs the cost of stage $i + 1$.

Implication for selective processing. Eq. 1 shows that false-positive activations incur computation even when they do not result in user-facing interventions, and that this cost accumulates as they propagate through the pipeline. This motivates progressive filtering in PRISM: lightweight semantic retrieval removes clearly irrelevant observations, while the contextual Verifier handles the remaining ambiguous candidates before response generation. Whether the additional verification stage is beneficial depends on the intervention rate, the filtering rates, and the relative costs of the pipeline stages. PRISM targets the regime in which interventions are sparse and a recall-oriented Retriever retains residual false-positive activations that would otherwise reach the more expensive Responder. Appendix A.2 analyzes this trade-off.

3 METHODS

Motivated by the false-positive computation overhead in Eq. 1, PRISM is designed to progressively suppress false alarms throughout its retrieve–verify–respond cascade. In this section, we describe not only how the overall system suppresses false-positive activations through selective processing, but also how each stage is individually designed to reduce the false alarm rate (FAR) at its own decision point while preserving potentially actionable observations.

3.1 SYSTEM OVERVIEW

At timestep t, PRISM receives visual and conversational observations $(x_t^{\text{vis}}, x_t^{\text{text}})$ together with an active user instruction r_t, such as "Please tell me when you find an apple." The overall pipeline is illustrated in Fig. 2. The first stage continuously uses lightweight encoders to identify observations that may be related to the instruction. For visual input, the Visual Retriever uses a lightweight vision–language encoder ψ_{V} to convert the instruction into a user-interest set and match each incoming frame against related visual concepts. For conversational input, State Retriever uses a lightweight text encoder ψ_{T} to retrieve semantic states describing 1) possible proactive responses, 2) situations in which the assistant should wait, and 3) the user's current situation. This stage only identifies potentially relevant candidates and does not decide whether the assistant should respond. To maintain high recall, we calibrate validation-set retrieval thresholds to keep the Stage 1 false-rejection rate below 10%; thresholds may differ across retrievers and state categories but remain fixed at test time. The retrieved candidates are then examined by the on-demand Verifier, a sub-1B language model, using the current multimodal context and recent history. Finally, only candidates that pass this contextual assessment are sent to the Responder, the final MLLM responsible for response generation.

3.2 STAGE 1: SEMANTIC RETRIEVAL

Description-based ranking retrieval. As shown in Fig. 2(a), the Visual Retriever identifies visual concepts relevant to the active instruction. Motivated by description-based visual recognition, which represents each concept using multiple prompt templates or natural-language descriptions (Radford et al., 2021; Menon & Vondrick, 2023; Pratt et al., 2023; Roth et al., 2023), we associate each concept c in a visual vocabulary $\mathcal{C}$ with a description set $\mathcal{D}(c) = \{d_{c,1}, \ldots, d_{c,N_c}\}$ and precompute their embeddings. The vocabulary used in our experiments is described in Appendix A.3. When the instruction changes, the Visual Retriever encodes it as $h_t^r = \text{norm}(\psi_{\text{V}}(r_t))$ and constructs an interest set $\mathcal{I}_t^{\text{V}} = \text{TopK}_c\{c : a_t(c) \geq \tau_c^{\text{enr}}\}$, where $a_t(c)$ is the instruction-to-concept similarity. This query-time operation determines which concepts are subsequently matched against incoming frames.

For each frame, the Visual Retriever computes $h_t^{\text{vis}} = \text{norm}(\psi_{\text{V}}(x_t^{\text{vis}}))$ and compares it with all description embeddings associated with concepts in $\mathcal{C}$. As illustrated in Fig. 2(b), we globally rank all descriptions and aggregate the strongest M ranks for each concept, rather than aggregating raw similarity scores as in prior description-based methods. Let $\pi_t(c, n)$ denote the rank of description $d_{c,n}$, where smaller values indicate stronger matches:

$$R_t^{\text{vis}}(c) = \frac{1}{M} \sum_{n \in \text{TopM}_{n'}\{-\pi_t(c,n')\}} \pi_t(c, n). \tag{2}$$

The frame prediction is $\widehat{\mathcal{C}}_t = \text{TopK}_c\{-R_t^{\text{vis}}(c)\}$, and the visual evidence passed downstream is $\mathcal{V}_t = \widehat{\mathcal{C}}_t \cap \mathcal{I}_t^{\text{V}}$. This rank-based aggregation is less sensitive to frame-dependent similarity scales and achieves a lower FAR than conventional mean aggregation in our evaluation (Fig. 4(c)). Visual Retriever remains a recall-oriented candidate filter and does not determine whether the requested condition has been satisfied. In practice, averaging a small number of the highest-ranked matches ($M \leq 10$) is sufficient across the evaluated settings.

Massive semantic-state retrieval. The lower path of Fig. 2(a) processes speech and direct queries using State Retriever. Its vocabulary is partitioned as $\mathcal{Q} = \mathcal{Q}^{\text{pro}} \cup \mathcal{Q}^{\text{pend}} \cup \mathcal{Q}^{\text{user}}$, where proactive states $\mathcal{Q}^{\text{pro}}$ describe potential assistance opportunities, pending states $\mathcal{Q}^{\text{pend}}$ indicate that intervention should be postponed, and user states $\mathcal{Q}^{\text{user}}$ encode context relevant to verification or response generation. The proactive and user states are handled by a shared relevance branch, while the pending

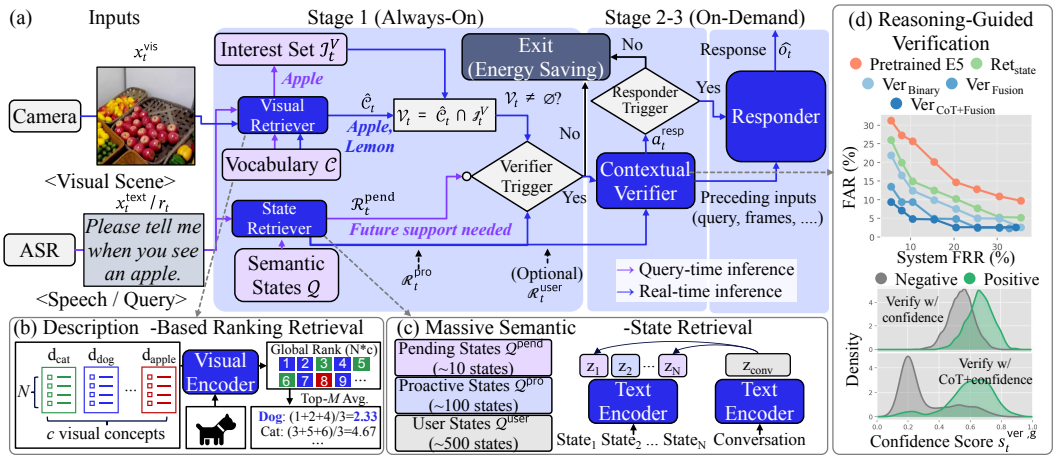


Figure 2: Overview of PRISM. (a) The three-stage cascade performs always-on semantic monitoring, on-demand contextual verification, and conditional answer generation. (b) Description-based ranking retrieval aggregates the top-M matched descriptions (M=3 in this example) for instruction-conditioned visual monitoring. (c) Multi-label contrastive learning aligns conversational observations with proactive, pending, and user semantic states. (d) The State Retriever evaluation compares hard binary verification with fusion of retrieval and Verifier scores, with and without explicit reasoning; the lower plots show the underlying Verifier-score distributions.

states are implemented as a separately trained gate over a compact anchor vocabulary. Representative examples are provided in Appendix Fig. 9.

Given a nonempty textual observation x_t^{text}, the State Retriever computes $h_t^{\text{text}} = \text{norm}(\psi_{\text{T}}(x_t^{\text{text}}))$. For each state $q_i^g \in \mathcal{Q}^g$, where $g \in \{\text{pro}, \text{pend}, \text{user}\}$, it precomputes $z_i^g = \text{norm}(\psi_{\text{T}}(q_i^g))$ and computes $s_{t,i}^g = (h_t^{\text{text}})^\top z_i^g$. When no new textual observation is available, all state scores are set to zero, yielding empty retrieved sets. The retrieved sets are

$$\mathcal{R}_t^g = \left\{ q_i^g \in \mathcal{Q}^g : s_{t,i}^g \geq \tau_i^g \right\}, \qquad g \in \{\text{pro}, \text{pend}, \text{user}\}, \tag{3}$$

where $\tau_i^g > 0$ is the validation-calibrated retrieval threshold for state q_i^g, whereas all pending states share a single threshold, i.e., $\tau_i^{\text{pend}} = \tau^{\text{pend}}$ for every i. When multiple states are retrieved, we use the state with the largest threshold margin, $s_{t,i}^g - \tau_i^g$, and omit its state index in the subsequent equations for brevity.

To reduce irrelevant state retrieval, we adapt the text encoder using low-rank adaptation (LoRA) (Hu et al., 2022) with the multi-label contrastive objective illustrated in Fig. 2(c). For each state group g,

$$\mathcal{L}_g = -\frac{1}{B} \sum_{n=1}^{B} \frac{1}{|\mathcal{Y}_n^g|} \sum_{q^+ \in \mathcal{Y}_n^g} \log \frac{\exp(s(x_n, q^+)/\gamma)}{\sum_{q \in \mathcal{Q}^g} \exp(s(x_n, q)/\gamma)}, \tag{4}$$

where B is the batch size, $\mathcal{Y}_n^g$ is the set of positive states assigned to observation x_n, and γ is the temperature. This objective aligns each conversation with all of its valid states while separating semantically similar non-positive states. This adaptation also reduces the FAR relative to the pretrained E5-Small (Wang et al., 2024) encoder, as shown in Fig. 2(d). Training-data construction and the evaluation protocol for the State Retriever are provided in Appendix A.5.

Verifier trigger. The outputs of the Visual and State Retrievers are jointly used to determine whether a candidate should be admitted to the Verifier. For automatic proactive activation, the Verifier admission decision is

$$a_t^{\text{ver}} = \mathbb{I}\left[(\mathcal{V}_t \neq \emptyset \vee \mathcal{R}_t^{\text{pro}} \neq \emptyset) \wedge \mathcal{R}_t^{\text{pend}} = \emptyset \right]. \tag{5}$$

Thus, visual evidence or at least one retrieved proactive state can activate the Verifier, while any retrieved pending state prevents unnecessary contextual reasoning.

Direct user requests and suppression. A direct user request r_t bypasses the gate in Eq. 5 and invokes the Verifier directly. Pending verification is used only on this path, allowing the response to

be deferred when future evidence is required. To avoid repeated activations within a single direct request, query-specific suppression decisions are cached for 5 seconds. Sensitivity to this window is reported in Table 8.

3.3 Stage 2: Contextual Verification

For each admitted candidate, the Verifier conditions on the active instruction r_t, the current observations $(x_t^{\text{text}}, x_t^{\text{vis}})$, the retrieved proactive, pending, and user states $(\mathcal{R}_t^{\text{pro}}, \mathcal{R}_t^{\text{pend}}, \mathcal{R}_t^{\text{user}})$, and a bounded recent multimodal context $\mathcal{H}_t$ containing a longer temporal window of visual frames and conversation (e.g., 30 seconds of video). Whereas Stage 1 identifies semantic relevance from the incoming observations, the Verifier uses this broader context to assess whether the requested proactive condition is satisfied and whether the available evidence supports continued deferral.

Reasoning-guided verification. A direct approach is to use the Verifier's `Yes`/`No` predictions as hard triggers. However, such binary verification only modestly reduces the FAR of the State Retriever in Fig. 2(d). For each $g \in \{\text{pro}, \text{pend}\}$, the Verifier assesses whether the retrieved hypothesis is supported by the available multimodal evidence before producing a constrained `Yes`/`No` response. Let $\ell_t^g(\cdot)$ denote the corresponding answer-token logits. We define the Verifier's confidence that the retrieved hypothesis is supported as

$$s_t^{\text{ver},g} = \frac{\exp(\ell_t^g(\texttt{Yes}))}{\exp(\ell_t^g(\texttt{Yes})) + \exp(\ell_t^g(\texttt{No}))}. \tag{6}$$

We fuse this verification score with the corresponding State Retriever score:

$$\tilde{s}_t^g = s_t^g + \lambda_{\text{ver}}^g \left(s_t^{\text{ver},g} - 0.5\right), \qquad g \in \{\text{pro}, \text{pend}\}, \tag{7}$$

where $\lambda_{\text{ver}}^{\text{pro}} = 0.4$ and $\lambda_{\text{ver}}^{\text{pend}} = 0.1$ by default. Fusion raises the score of a supported state and lowers that of an unsupported state; a higher pending-state score favors deferral. As shown in Fig. 2(d), score fusion reduces FAR more effectively than hard binary verification, and adding explicit reasoning reduces it further. Performance largely saturates with a modest reasoning budget, so we use 120 reasoning tokens by default; sensitivity to this budget is reported in Appendix Fig. 11.

In addition to the verification scores, the Verifier may produce a textual summary $\widehat{\mathcal{X}}_t$ of the multimodal evidence and its reasoning. This summary can optionally be forwarded to the Responder as auxiliary context; when it is not used, we set $\widehat{\mathcal{X}}_t = \varnothing$.

Responder trigger. The Responder is activated only when no verified pending evidence blocks the candidate and sufficient proactive or Verifier-supported visual evidence is available:

$$a_t^{\text{resp}} = \mathbb{I}\left[\tilde{s}_t^{\text{pend}} < \tau^{\text{pend}}\right] \cdot \mathbb{I}[\tilde{s}_t^{\text{pro}} \geq \tau^{\text{pro}} \vee (\mathcal{V}_t \neq \emptyset \wedge s_t^{\text{ver,pro}} \geq 0.5)]\,. \tag{8}$$

Thus, verified pending evidence suppresses the candidate, whereas sufficiently strong proactive evidence or Verifier-supported visual evidence activates the Responder. This contextual calibration filters candidates that are semantically relevant but not yet actionable, primarily reducing the conditional false-positive activation rate f_2 in Eq. 1.

3.4 Stage 3: Selective Response

Only candidates with $a_t^{\text{resp}} = 1$ reach the Responder. Given the available inputs from the preceding stages, we ask the Responder whether it should wait for further evidence before responding and obtain a final *wait* score $s_t^{\text{wait}} = P\Big(\texttt{Yes} \mid r_t, x_t^{\text{vis}}, x_t^{\text{text}}, \mathcal{R}_t^{\text{pro}}, \mathcal{R}_t^{\text{pend}}, \mathcal{R}_t^{\text{user}}, \widehat{\mathcal{X}}_t, \mathcal{H}_t\Big)$ and makes the final decision $\hat{y}_t = a_t^{\text{resp}} \mathbb{I}\big[s_t^{\text{wait}} < \tau^{\text{wait}}\big]$. When $\hat{y}_t = 1$, the Responder generates

$$\hat{o}_t = \text{Responder}\Big(r_t, x_t^{\text{vis}}, x_t^{\text{text}}, \mathcal{R}_t^{\text{pro}}, \mathcal{R}_t^{\text{pend}}, \mathcal{R}_t^{\text{user}}, \widehat{\mathcal{X}}_t, \mathcal{H}_t\Big)\,.$$

Otherwise, the system remains silent.

4 Experiments

We evaluate PRISM primarily on PhoStream (Lu et al., 2026), a streaming multimodal-assistance benchmark with *Instant*, *Backward*, and *Forward* examples requiring current, preceding, and future

Table 1: Evaluation results on PhoStream. For decoder-gated baselines, a gate model decides whether to generate a response at the current point in the stream. Alternative routing criteria are analyzed in Appendix A.7. Parameter counts include all stages, and energy is normalized to direct Qwen3.5-0.8B. ER, NR, and PC denote Early Response, No Response, and Partly Correct on Forward tasks.

Model	Total Param. ↓	Norm. energy ↓	Overall ↑	Evaluation score ↑			Forward task analysis (%)		
				Instant	Backward	Forward	ER ↓	NR ↓	PC ↑
Direct response baselines (following Lu et al. (2026))									
Qwen3.5	0.8B	1.00	23.8	49.0	46.3	0.0	100.0	0.0	0.0
	2B	1.58	27.7	56.4	54.8	0.0	99.9	0.0	0.1
	4B	2.24	30.0	61.1	58.8	0.2	99.4	0.0	0.6
	9B	3.36	34.6	67.2	65.1	3.3	86.4	2.2	11.4
Qwen3-VL	8B	3.16	36.3	66.1	64.1	7.7	81.3	5.7	12.9
	30B	6.13	33.4	65.5	62.1	3.0	93.3	0.9	5.8
Qwen2.5-Omni	7B	2.96	29.1	59.6	56.2	0.3	97.9	0.2	1.9
Qwen3-Omni	30B	6.13	29.7	59.6	58.3	0.7	98.3	0.1	1.5
Decoder-gated response (gate: Qwen3.5-0.8B; inspired by the two-stage design of Chen et al. (2025))									
Qwen3.5	4.8B	3.23	30.9	63.3	59.1	0.6	98.1	0.7	1.2
	9.8B	4.35	35.4	68.3	65.2	4.0	89.3	3.8	6.9
PRISM	4.8B	**1.00**	**40.0**	58.7	55.9	22.7	9.4	48.0	42.6

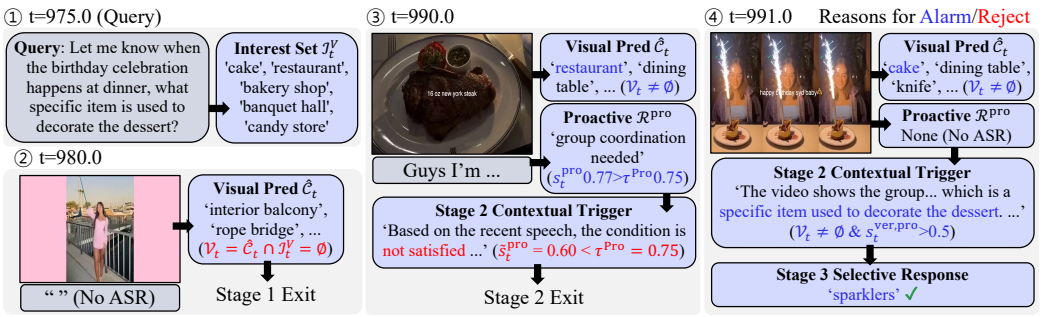


Figure 3: A Forward example in PhoStream illustrating the relevance–actionability gap. The Verifier rejects a relevant but insufficient observation at $t = 990$, and PRISM responds with "sparklers" at $t = 991$ after the requested evidence appears. Another example is in Fig. 10 in Appendix.

evidence, respectively. To ensure direct comparability with the original PhoStream study, we evaluate the same model families and baseline scope under its evaluation protocol, covering Qwen-family models (Team, 2026) including Omni variants (Xu et al., 2025a;b). We report response quality, intervention timing, estimated inference energy, and component-level false alarms. To assess whether the intermediate representations produced by PRISM also support reasoning without an explicit user query, we additionally evaluate on SIR-Bench.

Unless otherwise specified, PRISM uses E5-Small (Wang et al., 2024) as the State Retriever, PE-Core-Small (Bolya et al., 2025) as the Visual Retriever, Qwen3.5-0.8B as the Verifier, and Qwen3.5-4B as the Responder. Energy-estimation details are provided in Appendix A.1. Baselines and full implementation, training, and evaluation details are provided in Appendix A.3. Extended hyperparameter sweeps and sensitivity analyses of the final wait threshold, Verifier context window, reasoning-token budget, and temporal processing resolution are provided in Appendix A.4.

4.1 END-TO-END STREAMING PERFORMANCE

Task performance. Table 1 highlights intervention timing as the central challenge. Direct-response models struggle on *Forward* questions, and prompting a decoder to gate responses improves Forward scores by only 0.4 and 0.7 points over the corresponding direct Qwen3.5-4B and Qwen3.5-9B baselines, respectively. This suggests that deciding whether to intervene directly from raw inputs remains challenging for small decoder models (cf. Fig. 2(d)). In contrast, PRISM retrieves candidate evidence from the multimodal stream and verifies its actionability in context before responding. It

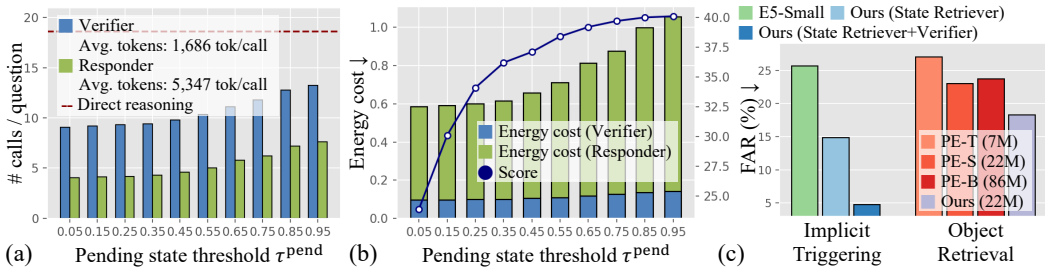


Figure 4: Efficiency and false-alarm analysis. (a) Verifier and Responder calls as τ^{pend} varies. (b) PhoStream score versus estimated normalized inference energy. (c) FAR for implicit triggering and zero-shot object retrieval on COCO (Lin et al., 2015). PE-T/S/B denote PE-Core-Tiny/Small/Base.

achieves the highest overall score (40.0 versus 36.3) and raises Forward from 7.7 to 22.7, while reducing Early Responses from 81.3% to 9.4%. Although this improved timing comes with more deferral than Qwen3-VL-8B (No Response: 5.7% → 48.0%) and modestly lower Instant and Backward scores than Qwen3.5-4B, reflecting false rejections from selective filtering, the higher overall score suggests that PRISM achieves a favorable trade-off.

Qualitative analysis. Fig. 3 makes this timing distinction concrete: the restaurant scene at $t = 990$ is relevant to the query but does not yet show the requested item, so the Verifier suppresses the candidate. When the item appears at $t = 991$, the Responder answers "sparklers."

4.2 Efficiency and Component Analysis

Performance–energy trade-off. Fig. 4(a–b) varies the pending-state threshold τ^{pend}. Raising this threshold relaxes early pending-state filtering, leaving more decisions to the Verifier and Responder. Over the default PhoStream monitoring horizon (18.63 processing ticks per question on average; Appendix A.1), Verifier and Responder calls per question increase gradually, with the score rising from 23.9 to 40.1 and normalized inference energy from 0.58 to 1.05. Even so, the operating point in Table 1 scores 40.0 at normalized energy 1.00, less than half that of Qwen3.5-4B (2.24).

Component-level false alarms. Fig. 4(c) shows that our components effectively reduce FAR at a target FRR of 10%. For implicit triggering, fine-tuning the State Retriever for massive semantic-state retrieval reduces FAR relative to the pretrained encoder, and contextual verification further suppresses false activations. For visual retrieval, the 22M-parameter rank-based Visual Retriever achieves a lower FAR than all evaluated raw-similarity baselines, including the 86M variant.

Verifier routing efficiency. Table 2 shows that, under extended monitoring (Appendix A.6), the Verifier reduces the Responder call rate from 36.9% to 10.2%, cutting inference energy by 2.7×. Separately, on the full PhoStream evaluation, we find that removing the Verifier improves the Overall score by approximately 1.15×: the 4B Responder can make the final decision on candidates otherwise rejected by the Verifier. Even this extended protocol spans only minutes given the benchmark's temporal coverage; we expect larger cumulative energy savings from selective routing over hours of everyday monitoring if candidate activations remain sparse.

Table 2: Stage-wise call rates for the Verifier routing ablation. Both configurations use the same Retriever and Stage 3 decision rule.

Routing	Ver.	Resp.	$E_{\text{rel}} \downarrow$
Full cascade	36.9%	10.2%	**1.0**×
w/o Verifier	–	36.9%	2.7×

4.3 Situational Intervention Reasoning

Motivation. Existing proactive-assistance benchmarks typically specify the assistance target through explicit instructions such as "Tell me when …" or "Wait until …". To evaluate intervention reasoning without such requests, we introduce the Situational Intervention Reasoning Benchmark (SIR-Bench). SIR-Bench requires models to infer an unstated user need from indirect conversational cues and ground an appropriate intervention in the surrounding visual context. Dataset construction and filtering details are provided in Appendix A.3.

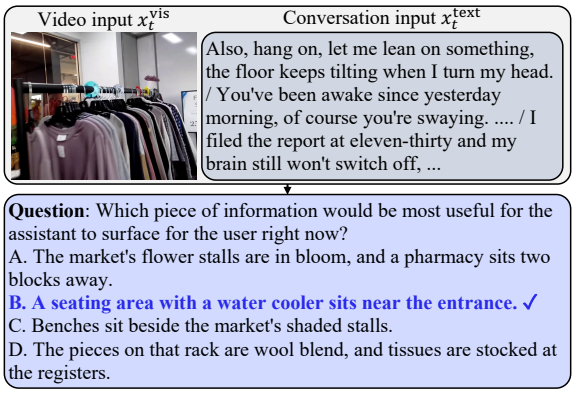


Figure 5: Example from SIR-Bench. The assistant infers the user's latent state from indirect visual and conversational cues and selects the most contextually appropriate response. More examples are in Fig. 8 in Appendix.

Method	x_t^{vis}	x_t^{text}	$\mathcal{R}_t^{\text{user}}$	$\hat{\mathcal{X}}_t$	Acc. ↑
Qwen3.5-0.8B	✓	✓	–	–	36.8%
Qwen3.5-2B	✓	✓	–	–	55.2%
Qwen3.5-4B	✓	✓	–	–	60.7%
Qwen3.5-9B	✓	✓	–	–	**70.1%**
PRISM	–	–	–	–	25.4%
	✓	–	–	–	37.3%
	–	✓	–	–	59.2%
	✓	✓	–	–	60.7%
	✓	✓	✓	–	62.2%
	✓	✓	–	✓	64.2%
	✓	✓	✓	✓	**65.2%**

Table 3: Raw-input baselines and PRISM input ablations on SIR-Bench. The no-input accuracy of 25.4% is close to the four-way chance level of 25.0%, suggesting little exploitable answer-only bias.

Setting. Fig. 5 illustrates the task. The user never directly asks for assistance, but the conversation suggests physical discomfort and fatigue, while the visual scene contains several possible resources. The model must infer the user's situation and determine which environmental information would be helpful—in this case, directing the user to a nearby seating area with water. Neither the assistance target nor the relevant visual evidence is specified in advance.

Results. Table 3 shows that raw-input accuracy increases from 36.8% with Qwen3.5-0.8B to 70.1% with Qwen3.5-9B. We further evaluate whether the intermediate representations of PRISM support such reasoning. With Qwen3.5-4B, retrieved user states and the Verifier summary progressively improve accuracy, and combining both reaches 65.2% without increasing the Responder size.

5 Related Works

Proactive agentic systems. Prior work studies proactive assistance through on-device context recognition and intervention learning (Khan & Maes, 2021; Lu et al., 2025; Liu et al., 2025; Zhang et al., 2025), context-, memory-, and planning-aware agents (Huang et al., 2025; Yang et al., 2025a; Pu et al., 2025; Sitong et al., 2026; Xie et al., 2026), and efficient streaming or task-progress modeling (Chen et al., 2025; Yu et al., 2025; Xu et al., 2026); complementary benchmarks evaluate proactive interaction quality, intent prediction, and timing (Wang et al., 2025; Ran et al., 2026). Across these directions, proactive assistance is largely framed as predicting or reasoning about whether and when to intervene. In contrast, PRISM uses lightweight retrieval to identify candidate moments and verifies their actionability before intervening.

Egocentric understanding. Prior work includes large-scale egocentric benchmarks for activity, daily-life, and cross-view understanding (Damen et al., 2022; Grauman et al., 2022; 2024); video–language representation learning (Lin et al., 2022; Zhao et al., 2023); and benchmarks for task reasoning, long-form understanding, and temporally grounded question answering (Jia et al., 2022; Mangalam et al., 2023; Di & Xie, 2024). Related studies address long-horizon assistance and structured video representation (Yang et al., 2025b; Rodin et al., 2024), as well as audio-visual understanding, mobile memory, personalized retrieval, and interaction prediction (Seth et al., 2026; Chen et al., 2026; Tang et al., 2026; Wang et al., 2026).

6 Conclusion

We presented PRISM, a false-alarm-aware retrieve–verify–respond framework for efficient always-on proactive assistance. By separating lightweight relevance retrieval from contextual actionability verification, PRISM reserves expensive reasoning for selected candidates. On PhoStream, it improves Forward-task performance, reduces early responses, and achieves a better performance–energy trade-off under our estimation model than direct MLLM baselines. SIR-Bench further shows that its intermediate representations support reasoning from implicit multimodal cues.

AI USE STATEMENT

Generative AI systems were used to construct SIR-Bench by synthesizing video-conditioned conversations and multiple-choice questions and filtering answer-only cases (details are in Appendix A.3); to generate, augment, and annotate the synthetic training and held-out evaluation data for the State Retriever, as detailed in Appendix A.5; to support model-based evaluation for PhoStream benchmark (Appendix A.3); and to polish the writing of this paper. All AI-generated data, annotations, evaluation outputs, and edited text were reviewed by the authors, who take full responsibility for the content of this work.

ETHICS STATEMENT

Always-on proactive assistance raises privacy, consent, and false-intervention risks. We collect no new human-subject data and use existing research datasets under their applicable terms; SIR-Bench combines Ego4D Grauman et al. (2022) clips with synthetically generated conversations and questions. Synthetic generation and automated annotation may introduce cultural, demographic, or situational biases, and our results should not be interpreted as evidence of safe real-world deployment. The system may miss useful interventions or respond unnecessarily and should not be used for medical, emergency, or other safety-critical decisions without additional validation and human oversight. Any deployment would require informed consent, data minimization, secure retention, and meaningful user control.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollar, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 60884–60937. Curran Associates, Inc., 2025. doi: 10.52202/085713-2036. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/57bc0a850255e2041341bf74c7e2b9fa-Paper-Conference.pdf.

Tuochao Chen, Nicholas Scott Batchelder, Alisa Liu, Noah A. Smith, and Shyamnath Gollakota. LlamaPIE: Proactive in-ear conversation assistants. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13801–13824, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.710. URL https://aclanthology.org/2025.findings-acl.710/.

Yijun Chen, Boyi Xiao, Yixian Zhao, Haoting Xia, Buqiang Xu, Jizhan Fang, Yanya Li, Yaqi Zheng, Xuehai Wang, Zirui Xue, Liuxin Zhang, Hui Li, and Ningyu Zhang. Lightmem-ego: Your ai memory for everyday life, 2026. URL https://arxiv.org/abs/2607.11487.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *Int. J. Comput.*

Vision, 130(1):33–55, January 2022. ISSN 0920-5691. doi: 10.1007/s11263-021-01531-2. URL https://doi.org/10.1007/s11263-021-01531-2.

Shangzhe Di and Weidi Xie. Grounded Question-Answering in Long Egocentric Videos . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12934–12943, Los Alamitos, CA, USA, June 2024. IEEE Computer Society. doi: 10.1109/CVPR52733.2024.01229. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01229.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, June 2022.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C.V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19383–19400, June 2024.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yifei Huang, Jilan Xu, Baoqi Pei, Lijin Yang, Mingfang Zhang, Yuping He, Guo Chen, Xinyuan Chen, Yaohui Wang, Zheng Nie, Jinyao Liu, Dechen Lin, Fang Fang, Kunpeng Li, Chang Yuan, Yu Qiao, Yali Wang, and Limin Wang. Vinci: A real-time smart assistant based on egocentric vision-language model for portable devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 9 (3), September 2025. doi: 10.1145/3749513. URL https://doi.org/10.1145/3749513.

Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. Egotaskqa: Understanding human tasks in egocentric videos. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh

(eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 3343–3360. Curran Associates, Inc., 2022. doi: 10.52202/068431-0242. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/161c94a58ca25bafcaf47893e8233deb-Paper-Datasets_and_Benchmarks.pdf.

Vaiva Kalnikaite, Abigail Sellen, Steve Whittaker, and Dave Kirk. Now let me see where i was: Understanding how lifelogs mediate memory. In *Proceedings of CHI 2010*. Association for Computing Machinery, Inc., April 2010. URL https://www.microsoft.com/en-us/research/publication/now-let-me-see-where-i-was-understanding-how-lifelogs-mediate-memory/.

Mina Khan and Pattie Maes. Pal: Intelligence augmentation using egocentric visual context detection, 2021. URL https://arxiv.org/abs/2105.10735.

Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. SODA: Million-scale dialogue distillation with social commonsense contextualization. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12930–12949, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.799. URL https://aclanthology.org/2023.emnlp-main.799/.

Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z. XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, WANG HongFa, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 7575–7586. Curran Associates, Inc., 2022. doi: 10.52202/068431-0550. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/31fb284a0aaaad837d2930a610cd5e50-Paper-Conference.pdf.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

Xingyu Bruce Liu, Shitao Fang, Weiyan Shi, Chien-Sheng Wu, Takeo Igarashi, and Xiang 'Anthony' Chen. Proactive conversational agents with inner thoughts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713941. doi: 10.1145/3706598.3713760. URL https://doi.org/10.1145/3706598.3713760.

Xudong Lu, Guan Huankang, Yang Bo, Jinpeng Chen, Xintong Guo, Shuhan LI, Fang Liu, Peiwen Sun, Xueying Li, Wei Zhang, Xue Yang, Rui Liu, and Hongsheng Li. Phostream: Benchmarking real-world streaming for omnimodal assistants in mobile scenarios. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=cCMbxJcDmH.

Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, Weiwen Liu, Yasheng Wang, Zhiyuan Liu, Fangming Liu, and Maosong Sun. Proactive agent: Shifting LLM agents from reactive responses to active assistance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sRIU6k2TcU.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46212–46244. Curran Associates, Inc., 2023. doi: 10.52202/075280-2004. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/90ce332aff156b910b002ce4e6880dec-Paper-Datasets_and_Benchmarks.pdf.

Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jlAjNL8z5cs.

Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15645–15655, 2023. doi: 10.1109/ICCV51070.2023.01438.

Kevin Pu, Ting Zhang, Naveen Sendhilnathan, Sebastian Freitag, Raj Sodhi, and Tanya R. Jonker. Promemassist: Exploring timely proactive assistance through working memory modeling in multi-modal wearable devices. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720376. doi: 10.1145/3746059.3747770. URL https://doi.org/10.1145/3746059.3747770.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Dongchuan Ran, Linyu Ou, Xueheng Li, Wenwen Tong, Chenxu Guo, Hewei Guo, Kaibing Wang, and Lewei Lu. Egopro-bench: Benchmarking personalized proactive interaction in egocentric video streams, 2026. URL https://arxiv.org/abs/2605.07299.

Ivan Rodin, Antonino Furnari, Kyle Min, Subarna Tripathi, and Giovanni Maria Farinella. Action scene graphs for long-form understanding of egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18622–18632, 2024.

Karsten Roth, Jae Myung Kim, A. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15746–15757, October 2023.

Samsung Electronics. Samsung Galaxy S26, 2026. URL https://www.samsung.com/sec/smartphones/galaxy-s26/.

Ashish Seth, Xinhao Mei, Changsheng Zhao, Varun Nagaraja, Ernie Chang, Gregory P. Meyer, Gael Le Lan, Yunyang Xiong, Vikas Chandra, Yangyang Shi, Dinesh Manocha, and Zhipeng Cai. Egoavu: Egocentric audio-visual understanding, 2026. URL https://arxiv.org/abs/2602.06139.

Gong Sitong, Tianyu Yan, Caixin Kang, Bo Zheng, Xiang Ruan, Huchuan Lu, Kaipeng Zhang, Yoichi Sato, and Yifei Huang. Vinci2: Providing proactive assistance in continuous egocentric videos, 2026. URL https://arxiv.org/abs/2607.11523.

Yuanmin Tang, Jue Zhang, Xiaoting Qin, Jing Yu, Meikang Qiu, Gaopeng Gou, Gang Xiong, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Wu. EgoMemory: Memory-augmented personalized retrieval for long-context egocentric video. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7317–7349, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.362. URL https://aclanthology.org/2026.findings-acl.362/.

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

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

Yanshuo Wang, Yuan Xu, Xuesong Li, Jie Hong, Yizhou Wang, Chang Wen Chen, and Wentao Zhu. Egoself: From memory to personalized egocentric assistant, 2026. URL https://arxiv.org/abs/2604.19564.

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

Zhifei Xie, Zongzheng Hu, Fangda Ye, Xin Zhang, Haobo Chai, Zihang Liu, Pengcheng Wu, Guibin Zhang, Yue Liao, Xiaobin Hu, Deheng Ye, Chunyan Miao, and Shuicheng Yan. Pask: Toward intent-aware proactive agents with long-term memory, 2026. URL https://arxiv.org/abs/2604.08000.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025a. URL https://arxiv.org/abs/2503.20215.

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

Lilin Xu, Bufang Yang, Siyang Jiang, Kaiwei Liu, Kaiyuan Hou, Yuang Fan, Hongkai Chen, Zhenyu Yan, and Xiaofan Jiang. Pro^{2}assist: Continuous step-aware proactive assistance with multi-modal egocentric perception for long-horizon procedural tasks, 2026. URL https://arxiv.org/abs/2605.04227.

Bufang Yang, Lilin Xu, Liekang Zeng, Kaiwei Liu, Siyang Jiang, Wenrui Lu, Hongkai Chen, Xiaofan Jiang, Guoliang Xing, and Zhenyu Yan. Contextagent: Context-aware proactive LLM agents with open-world sensory perceptions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=tRXt10xKc5.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, Bei Ouyang, Zhengyu Lin, Marco Cominelli, Zhongang Cai, Bo Li, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, Lei Yang, and Ziwei Liu. Egolife: Towards egocentric life assistant. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28885–28900, June 2025b.

Xueyang Yu, Cheng Shi, Yang Wang, and Sibei Yang. Eyes wide open: Ego proactive video-llm for streaming video. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 13420–13463. Curran Associates, Inc., 2025. doi: 10.52202/085713-0453. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/141304a37d59ec7f116f3535f1b74bde-Paper-Conference.pdf.

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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12044–12068, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.605. URL https://aclanthology.org/2025.emnlp-main.605/.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6586–6597, 2023. doi: 10.1109/CVPR52729.2023.00637.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

A Appendix

A.1 Energy Model

We estimate model-inference energy on a Samsung Galaxy S26 (Samsung Electronics, 2026) using NPU prefill-throughput profiles. The purpose of this analysis is to compare inference configurations under a common hardware model, rather than to estimate end-to-end battery consumption.

Per-call prefill energy. For each model m and input length L, we convert the profiled prefill latency $\hat{t}_m^{\mathrm{pre}}(L)$ into an energy estimate as

$$\widehat{E}_m^{\mathrm{pre}}(L) = \hat{t}_m^{\mathrm{pre}}(L) P_{\mathrm{NPU}}, \tag{9}$$

where we use a fixed NPU power assumption of $P_{\mathrm{NPU}} = 1{,}750\,\mathrm{mW}$. With latency measured in seconds, $\widehat{E}_m^{\mathrm{pre}}(L)$ is reported in mJ. Because the same power assumption is applied to all models, these values should be interpreted as relative model-inference estimates rather than direct device-level energy measurements.

Fig. 6 shows the resulting relative prefill-energy profiles at input lengths of 128, 256, 512, 1,024, 2,048, and 4,096 tokens. We profile the lightweight E5-Small and PE-Core-S16-384 encoders together with Qwen3-0.6B and Qwen3-4B, and use an estimated Qwen3-30B profile for large-model comparison. All values are normalized by the estimated prefill energy of E5-Small at 128 input tokens, which is assigned a value of 1.0. The profiles show a substantial cost gap between lightweight encoders and decoder-based reasoning models, motivating the use of inexpensive monitoring before invoking downstream language models.

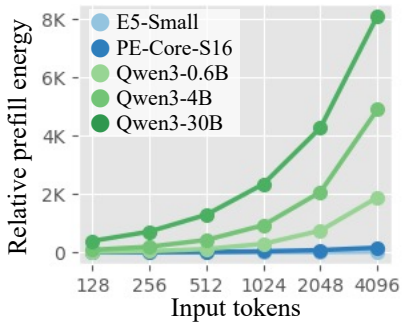


Figure 6: Relative prefill energy across input lengths. All values are normalized by the estimated prefill energy of E5-Small at 128 input tokens. The estimates are derived from NPU prefill profiles under a fixed 1.75 W power assumption.

Model proxies and observed workload. PRISM uses Qwen3.5-0.8B as the Verifier and Qwen3.5-4B as the Responder. Because matching Galaxy S26 profiles are unavailable for these exact checkpoints, we use Qwen3-0.6B and Qwen3-4B, respectively, as the closest available energy proxies. We measure Verifier and Responder prompt lengths from actual model calls over 606 questions across randomly selected 45 PhoStream videos. The mean input lengths are 1,686.1 tokens for the Verifier and 5,347.7 tokens for the Responder.

Separately, a PhoStream question contains an average of 18.63 candidate timesteps under our 1-s evaluation stride. An ungated reasoning model is therefore invoked 18.63 times per question on average. The lightweight monitor is evaluated throughout this stream, whereas PRISM invokes the Verifier and Responder only on candidates admitted by the preceding stages; their invocation counts are measured directly for each operating point.

Interpolation at observed context lengths. The observed prompt lengths generally do not coincide with the profiled input lengths. For $L_1 \le L \le L_2$, we linearly interpolate between the neighboring profile points:

$$\widehat{E}_m^{\mathrm{pre}}(L) = E_{m,1} + \frac{L - L_1}{L_2 - L_1}\left(E_{m,2} - E_{m,1}\right), \tag{10}$$

where $E_{m,1} = \widehat{E}_m^{\mathrm{pre}}(L_1)$ and $E_{m,2} = \widehat{E}_m^{\mathrm{pre}}(L_2)$. For prompts longer than the largest profiled length of 4,096 tokens, we linearly extrapolate using the two largest profile points.

Applying this procedure to the observed mean prompt lengths gives approximately 590 mJ per Verifier call and 6,628 mJ per Responder call. Thus, under the observed workload, one Responder invocation has approximately

$$\frac{\widehat{E}_{\mathrm{Resp}}^{\mathrm{pre}}}{\widehat{E}_{\mathrm{Ver}}^{\mathrm{pre}}} \approx 11.2 \tag{11}$$

times the estimated prefill energy of one Verifier invocation.

Question-level inference cost. Let $\bar{T} = 18.63$ denote the average number of candidate timesteps per question, and let $N_{\mathrm{Ver}}(\tau)$ and $N_{\mathrm{Resp}}(\tau)$ denote the measured average numbers of Verifier and Responder invocations at operating point τ. We estimate the average inference cost of PRISM as

$$\widehat{E}_{\mathrm{PRISM}}(\tau) = \bar{T}\widehat{E}_{\mathrm{Ret}} + N_{\mathrm{Ver}}(\tau)\widehat{E}^{\mathrm{pre}}_{\mathrm{Ver}} + N_{\mathrm{Resp}}(\tau)\widehat{E}^{\mathrm{pre}}_{\mathrm{Resp}}, \tag{12}$$

where $\widehat{E}_{\mathrm{Ret}}$ denotes the per-timestep inference cost of the State and Visual Retrievers. For direct reasoning with model m, where the model is invoked at every candidate timestep, the corresponding estimate is

$$\widehat{E}_{\mathrm{direct},m} = \bar{T}\,\widehat{E}^{\mathrm{pre}}_{m}. \tag{13}$$

Thus, differences between operating points arise primarily from how many candidates are propagated to the substantially more expensive Verifier and Responder stages.

For the normalized energy values reported in the main experiments, we divide each question-level estimate by the corresponding estimate for the ungated Qwen3.5-0.8B baseline, which is assigned a value of 1.0. This normalization is used only to facilitate relative comparison across inference configurations.

Scope, empirical basis, and interpretation. Our analysis is designed to compare model-routing configurations under a common mobile-hardware setting rather than to predict end-to-end battery life. It is grounded in two measured quantities: prefill-latency profiles collected on the Galaxy S26 NPU across model sizes and input lengths, and the prompt lengths and stage-invocation counts observed over the full PhoStream evaluation. The modeled components are limited to converting latency into energy under a shared power assumption, interpolating or extrapolating between profiled input lengths, and using the closest available Qwen3 checkpoints as proxies for the corresponding Qwen3.5 models. Because the same P_{NPU} is applied to every stage and configuration, it cancels in the normalized comparisons and affects only the absolute mJ scale. We restrict the accounting to prefill because comparable on-device decode profiles are unavailable across all evaluated model sizes, avoiding the introduction of additional unmatched assumptions. Accordingly, our conclusions concern relative model-inference cost and are driven primarily by the profile-derived per-call cost differences and the measured routing frequencies; the reported values should not be interpreted as direct measurements of total device energy or battery drain.

A.2 Architecture Selection under Sparse Intervention Events

The main experiments show that contextual verification reduces Responder calls under extended monitoring. Here, we analyze when direct reasoning, a two-stage Retriever–Responder pipeline, or a three-stage Retriever–Verifier–Responder pipeline minimizes expected inference energy. This analysis is intended to illustrate the architectural trade-off rather than reproduce the exact PhoStream workload.

Analysis setup. To control for context length, we fix the input to 4,096 tokens for all stages and use the corresponding estimates from Appendix A.1. The Retriever cost e_{Ret} combines the State Retriever (E5-Small) and Visual Retriever (PE-Core-S16), while e_{Ver} and e_{Resp} use the Qwen3-0.6B and Qwen3-4B profiles, respectively. The resulting per-observation costs are approximately 198.9 mJ, 1,887.0 mJ, and 4,898.4 mJ.

Let p denote the probability that an observation requires intervention, and let f_1 denote the conditional probability that the Retriever forwards an observation that does not require intervention. We assume $\rho_1 = \rho_2 = 1$, such that observations requiring intervention are retained by both filtering stages. The analysis therefore isolates computation caused by observations that do not require intervention. For illustration, we set the Verifier's conditional false-positive activation rate to

$$f_2 = \frac{1}{2} f_1. \tag{14}$$

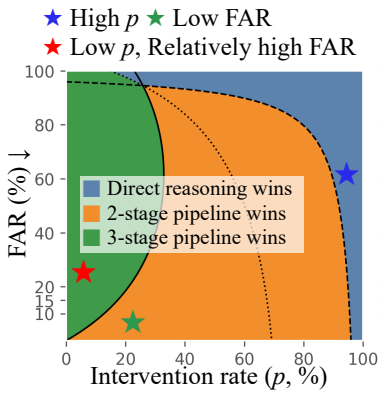


Figure 7: Energy-preferred architecture as a function of the intervention rate p and Stage 1 false-positive activation rate f_1. All stage costs are evaluated at a fixed input length of 4,096 tokens.

This assumption determines the precise regime boundaries in Fig. 7, but the qualitative trade-off depends only on whether the Responder calls avoided by verification outweigh the additional Verifier cost.

Under these assumptions, the expected inference energy per observation is

$$E_{\text{direct}} = e_{\text{Resp}}, \tag{15}$$

$$E_{\text{2stage}} = e_{\text{Ret}} + [p + (1-p)f_1]\, e_{\text{Resp}}, \tag{16}$$

$$E_{\text{3stage}} = e_{\text{Ret}} + [p + (1-p)f_1]\, e_{\text{Ver}} + [p + (1-p)f_1 f_2]\, e_{\text{Resp}}. \tag{17}$$

For each (p, f_1) pair, Fig. 7 reports the architecture with the lowest expected energy.

Architecture regimes. When interventions are frequent, most observations require downstream reasoning, leaving few Responder calls for the filtering stages to avoid. The additional stage overhead can therefore make direct reasoning preferable. As interventions become sparser, early retrieval becomes increasingly useful. If the Retriever already achieves a low f_1, the two-stage architecture captures most of the available savings, and an additional Verifier provides limited benefit. In contrast, when interventions are sparse but the recall-oriented Retriever retains residual false activations, these activations accumulate over time. The three-stage architecture is then preferred whenever the Verifier prevents enough expensive Responder calls to offset its own cost.

Implication for PRISM. The analysis shows that a deeper cascade is not inherently more efficient: its value depends on intervention sparsity, upstream selectivity, and relative stage costs. PRISM targets the regime in which intervention-requiring events are rare but the Retriever intentionally retains ambiguous candidates to preserve recall. The Verifier removes these residual activations before the substantially more expensive Responder. This prediction is consistent with Table 2, which shows substantial energy savings from verification under extended monitoring.

A.3 EXPERIMENTAL DETAILS

Models. We evaluate language and multimodal models ranging from 0.8B to 30B parameters. The end-to-end baselines include Qwen3-VL-8B, Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025), Qwen2.5-Omni-7B (Xu et al., 2025a), and Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025b). We additionally evaluate the Qwen3.5 family (Team, 2026) at 0.8B, 2B, 4B, and 9B scales to characterize how proactive decision quality changes with model size.

The State Retriever uses E5-Small, a 22M-parameter text encoder, to retrieve proactive, pending, and user states from ASR transcripts. We compare the pretrained encoder with its LoRA-adapted variant trained using the multi-label contrastive objective in Sec. 3.2. For the Visual Retriever, we evaluate three lightweight encoder configurations: PE-Core-Tiny (7M), PE-Core-Small (22M), and PE-Core-Base (86M). Unless otherwise stated, the Verifier uses Qwen3.5-0.8B and the Responder uses Qwen3.5-4B.

PhoStream details. We use PhoStream for end-to-end evaluation of proactive multimodal assistance. PhoStream divides its evaluation into *Instant*, *Backward*, and *Forward* categories. Instant examples can be resolved from the current observation, Backward examples require information from preceding observations, and Forward examples require the agent to wait for a future cue before responding. To provide conversational observations x_t^{text} to the State Retriever, we transcribe the dialogue in each video using Qwen3-ASR-1.7B as an offline preprocessing step. To reflect the resource constraints of always-on, battery-powered deployment, we downsample all FHD video inputs to 320×240 resolution and apply the same preprocessing across all evaluations. For evaluation, we employ Qwen3.5-122B-A10B as the LLM judge to assess response quality and proactive decision outcomes across different model configurations. For the main PhoStream experiments, the retrieved states and optional Verifier summary are used only for upstream routing and are not provided to the Responder; in the Responder input, we set $\mathcal{R}_t^{\text{pro}} = \mathcal{R}_t^{\text{pend}} = \mathcal{R}_t^{\text{user}} = \widehat{\mathcal{X}}_t = \varnothing$. Following the original PhoStream evaluation protocol, for Forward questions, we use a 2-second response window and run inference at six time points: Timestamp Question and Timestamp Proactive, each with offsets of 0, +1, and +2 seconds.

More details on SIR-Bench. We construct SIR-Bench from 22 randomly sampled 30-second video clips from Ego4D (Grauman et al., 2022). Each video clip is used without modification, while Claude Sonnet 5 synthesizes an accompanying conversation transcript and four-choice questions that

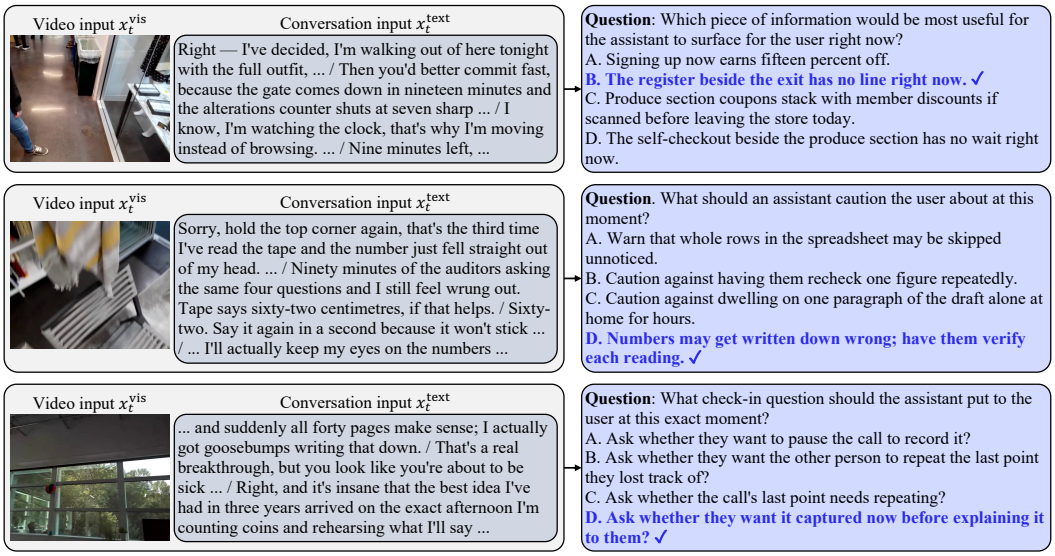


Figure 8: Representative examples from SIR-Bench requiring joint reasoning over video and conversation inputs.

Example of Proactive States
"decision or choice support needed", "financial calculation or decision needed", "academic or study support needed", "navigation or transit lookup needed", "scheduling or calendar action needed", ...

Example of Pending States
"explicit hold until a specified time or event", "action gated behind a specific condition", "task deferred to a later time", "no intervention requested on a particular matter", "higher priority task superseding current matter", ...

Example of User States
"driving a vehicle", "taking prescription medication", "reading e-books or digital documents", "urgently needing help nearby", "needs someone to intervene now", "technology, software, or computers", ...

Figure 9: Representative semantic state descriptions used for monitoring. Proactive states capture situations where assistance may be useful (e.g., decision support or scheduling needs); pending states encode deferred intentions and unmet conditions; user states describe persistent contextual attributes and ongoing activities. Rather than relying on task-specific labels, PRISM monitors observations through retrievable natural-language state descriptions.

require joint reasoning over both modalities. The video specifies the user's environment and ongoing activity, whereas the synthetic conversation conveys implicit factors such as urgency, cognitive state, emotional state, or temporal constraints. To reduce answer-only artifacts, we present each candidate question to Claude Sonnet 5 without either the video or conversation and remove questions that it answers correctly. After filtering, SIR-Bench contains 201 multiple-choice questions. Fig. 8 presents representative examples. On SIR-Bench, $\mathcal{R}_t^{\text{pro}}$ and $\mathcal{R}_t^{\text{pend}}$ remain empty, while Table 3 evaluates $\mathcal{R}_t^{\text{user}}$ and $\widehat{\mathcal{X}}_t$ as optional auxiliary inputs to the Responder. Consistent with this filtering procedure, the no-input accuracy is 0.254, close to the four-way chance level of 0.25.

Examples of semantic states. To support instruction-conditioned monitoring, we represent persistent user context using semantic state descriptions. Fig. 9 shows representative examples from the three semantic state categories introduced in Sec. 3.2: proactive states, pending states, and user states. These descriptions serve as high-level semantic abstractions that allow observations with different surface forms to be mapped to a shared state representation.

Visual encoder evaluation and vocabulary. The visual concept vocabulary $\mathcal{C}$ is configurable and is not tied to a fixed taxonomy. In our experiments, we construct $\mathcal{C}$ from the place categories in Places365 (Zhou et al., 2017) and the object categories in COCO (Lin et al., 2015). Accordingly, we use Places365 for place recognition and COCO for object recognition in the visual encoder evaluation.

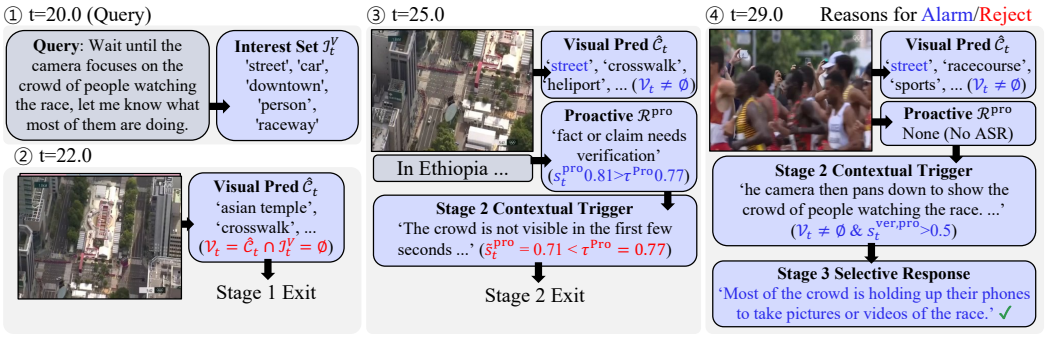


Figure 10: Another Forward example in PhoStream illustrating the relevance–actionability gap. The Verifier rejects a relevant but insufficient observation at $t = 25$, and PRISM responds correctly at $t = 29$ once the requested crowd becomes visible.

Table 4: Extended sweep of the pending-state threshold τ^{pend}. Increasing τ^{pend} makes filtering less aggressive, recovering more Instant and Backward examples while leaving Forward performance largely stable.

τ^{pend}	Overall ↑	Instant ↑	Backward ↑	Forward ↑
0.05	23.9	25.6	20.6	24.3
0.15	30.1	38.9	33.3	23.6
0.25	34.1	46.7	43.0	23.2
0.35	36.2	51.5	47.9	22.6
0.45	37.1	53.2	50.5	22.2
0.55	38.4	56.0	53.1	22.2
0.65	39.2	57.5	54.7	22.3
0.75	39.7	58.3	55.7	22.4
0.85	40.0	58.7	55.9	22.7
0.95	40.1	58.9	55.9	22.7

A.4 ADDITIONAL RESULTS

Unless otherwise specified, the following sensitivity analyses vary one system parameter around a common diagnostic reference configuration, whose Overall score is 37.1.

Additional qualitative example in PhoStream. Fig. 10 shows another Forward example in which semantic relevance precedes actionability. At $t = 25$, the Verifier suppresses the candidate because the requested crowd is not yet visible; once the crowd appears at $t = 29$, PRISM correctly reports that most people are holding up their phones to take pictures or videos.

Verifier reasoning length. Fig. 11 examines the effect of the Verifier's reasoning-token budget using the area under the receiver operating characteristic curve (AUC), which measures how well the verification score separates candidates that warrant intervention from those that do not across decision thresholds. AUC improves rapidly with short reasoning traces and largely saturates beyond approximately 100 tokens. We therefore use a compact budget of 120 tokens by default, which lies within this saturated regime while avoiding unnecessarily long reasoning sequences.

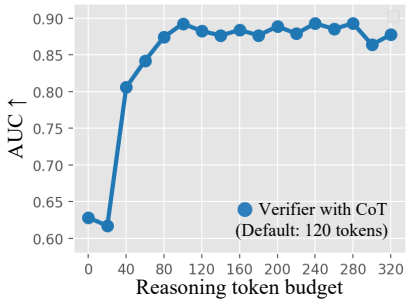


Figure 11: Verifier AUC as a function of the reasoning-token budget.

Pending-state threshold sensitivity. Table 4 provides an extended sweep of the same pending-state threshold τ^{pend} analyzed in Fig. 4. As τ^{pend} increases, pending-state filtering becomes less aggressive and more candidates are forwarded to downstream reasoning. Accordingly, Instant and Backward performance increases

Table 5: Sensitivity to the final wait threshold τ^{wait}.

τ^{wait}	Overall	Forward	Instant	Backward
0.3	36.3	24.6	49.3	46.5
0.4	37.2	23.4	52.3	49.9
0.5 (reference)	37.1	22.2	53.2	50.5
0.6	37.4	20.7	55.6	52.7
0.7	36.9	19.2	56.1	52.8
0.8	36.9	18.0	57.5	53.8

Table 6: Sensitivity to the trailing context window. All reported settings satisfy the multimodal input limit.

Window (s)	Overall	Forward	Instant	Backward
20	35.8	19.0	54.8	49.9
40	37.0	21.3	54.6	50.4
60 (reference)	37.1	22.2	53.2	50.5
90	36.6	20.4	54.7	52.9

substantially, from 25.6 to 58.9 and from 20.6 to 55.9, respectively. In contrast, Forward performance remains relatively stable between 22.2 and 24.3 across the sweep.

This behavior exposes the main selectivity trade-off of the cascade. Aggressive filtering reduces downstream computation but introduces false rejections, primarily affecting examples whose answers are already available from current or past context. Forward performance is considerably less sensitive to τ^{pend}, since these examples primarily depend on withholding a response until future evidence becomes available. Performance gains diminish beyond approximately $\tau^{\text{pend}} = 0.65$; we use $\tau^{\text{pend}} = 0.85$ for the main evaluation.

Final wait threshold. The threshold τ^{wait} controls the Responder's final decision to wait after a candidate reaches Stage 3. Because the Responder produces a response when $s_t^{\text{wait}} < \tau^{\text{wait}}$, increasing the threshold permits more responses. As shown in Table 5, this improves Instant and Backward performance while gradually reducing Forward performance. Nevertheless, the overall score remains within 0.8 points of the reference across the full range, and the best setting improves it by only 0.3. We therefore retain the default threshold of 0.5 rather than selecting a narrowly tuned operating point.

Verifier context window. We next vary the trailing transcript and session history available to the Verifier and Responder. As shown in Table 6, performance is stable between 40 and 90 seconds: the overall score varies by at most 0.5 points relative to the 60-second reference. The shorter 20-second window performs somewhat worse, particularly on Forward examples, indicating that a moderate amount of recent history is useful. We use 60 seconds in the reported configuration.

Temporal processing resolution. Finally, we vary the interval between consecutive processing ticks. A 0.5-second interval approximately doubles the number of evaluated ticks but improves the overall score by only 0.6, whereas a 2-second interval approximately halves the tick-level computation but reduces overall performance by 1.5, mainly through lower Forward performance. Table 7 therefore supports the 1-second interval as a practical accuracy–compute operating point.

Suppression persistence window. We vary how long a query-specific suppression remains cached before the system re-evaluates its routing decision. Table 8 reports the resulting sensitivity.

Without persistence, repeated re-evaluation substantially degrades Forward performance, whereas overly long windows delay reconsideration when new evidence becomes available. Moderate windows of 2–5 s provide the best trade-off and show stable overall scores. We use the fixed 5 s setting in the main experiments.

Table 7: Sensitivity to temporal processing resolution. Relative cost approximates the change in the number of evaluated ticks.

Interval (s)	Overall	Forward	Instant	Backward	Relative cost
0.5	37.7	22.5	54.1	51.4	$\sim 2.0\times$
1.0 (reference)	37.1	22.2	53.2	50.5	$1.0\times$
2.0	35.6	18.1	54.3	52.3	$\sim 0.5\times$

Table 8: Sensitivity to the suppression persistence window. Δ denotes the change in Overall score relative to the 5 s setting used in the main experiments.

Window (s)	Overall	Δ	Forward	Instant	Backward
0.0	29.7	-7.4	6.4	54.6	51.6
2.0	38.5	$+1.4$	24.2	54.2	51.5
3.0	38.3	$+1.2$	23.6	54.2	51.5
5.0	37.1	0.0	22.2	53.2	50.5
8.0	35.9	-1.2	18.9	54.2	51.5
12.0	32.8	-4.3	12.7	54.2	51.5

A.5 State Retriever Training Data and Evaluation

Existing dialogue benchmarks do not evaluate whether an assistant can identify situations that require proactive assistance when the user provides no explicit cue or request. To address this gap, we construct a new benchmark for *proactive implicit intervention* to evaluate if our State Retriever shows a reasonable level of performance. In this setting, the assistant must determine from an indirect transcription whether intervention is necessary, even though the user never explicitly asks for help. We construct both the training and test sets specifically for this purpose using conversations generated by a cloud LLM and conversations derived from SODA (Kim et al., 2023). For the conversations generated from scratch, we create scenarios in which an implicit need for assistance arises naturally during the dialogue. For the SODA-derived conversations, we augment the original dialogues with additional utterances that introduce a latent opportunity for assistance without explicitly requesting it. These examples form the positive samples, which are the conversations that contain no direct intervention cue, but an assistant should nevertheless respond proactively. We also construct negative examples consisting of conversations that do not require such AI assistance.

Each positive transcript is associated with two complementary label families rather than a single action label. *Proactive-action* labels specify the concrete assistance that the assistant should provide, such as *suggest setting a wake-up alarm*. In contrast, *user-state* labels describe the broader situational, affective, or activity-related condition that motivates the intervention, such as *being stressed or overwhelmed*. The State Retriever retrieves both label types so that the system can identify not only *what* assistance may be appropriate, but also the user's current state that provides the context for that assistance. In addition, we introduce *pending* states to represent situations in which the system should defer intervention because the relevant assistance is already pending or the current context does not yet warrant a new proactive response. Pending states therefore act as a suppression mechanism rather than as additional proactive actions.

We use the same construction procedure to create the training data for our State Retriever. Positive training transcripts are obtained from conversations generated from scratch or by rewriting dialogues from SODA, while transcripts with other proactive states work as in-batch negative examples. We fine-tune an E5-small encoder with LoRA applied to the query, key, and value projections for 5 epochs, using a batch size of 64, a linear learning-rate schedule, and a peak learning rate of 5×10^{-4}. Training uses the symmetric multi-positive InfoNCE (Information Noise-Contrastive Estimation) objective described in Sec. 3.2. We compute the objective separately for the 100 proactive-action labels and the 514 user-state labels. For pending-state handling, we additionally train a lightweight projection on top of a frozen copy of the fine-tuned encoder. This projection is trained using proactive and user-state representations together with pending-state anchors, and is used to determine whether a transcript corresponds to a condition in which the system should defer intervention. At inference time, the pending-state score is compared against a calibrated threshold. If the threshold is exceeded,

Positive sample 1

My train leaves at six tomorrow morning and I haven't even thought about setting an alarm.
Six in the morning? That's brutal, especially after being on the road all day.
I know — I'm such a heavy sleeper, I could easily sleep right through it.
Missing that train would be a real mess too, there's no later one that works.
Yeah, I really can't afford to oversleep.

Positive sample 2

We're heading out for the hike in about an hour and the sky already looks pretty overcast — no idea if it's going to hold or turn into a full-on storm.
Yeah, and it feels way more humid than I expected for this time of year.
The trail gets really muddy when it rains, and we'd be stuck out there with no shelter for miles.
I just wish there was a clearer picture of what the weather's doing this afternoon.
Last thing I want is to be halfway up the ridge when it decides to pour.

Negative sample 1

Hi, friend! How are you doing today?
I'm doing well, thank you. How about you?
I'm great, thanks. Just enjoying my favorite CD on repeat as usual.
Oh, which CD is that?
It's called “The Marshall Mathers LP” by Eminem.
I see. I don't think I've ever heard of that one.
Yeah, it's not really mainstream, but I love it. It's so raw and real.
It sounds like your kind of music. Do you like all of his songs?
Yeah, pretty much. He's a great rapper and storyteller. His lyrics are amazing.

Negative sample 2

Hi, thanks for delivering my pizza!
No problem, enjoy!
Yeah, I love pepperoni and pineapple together.
Oh, yeah? I've never tried that combination before.
Really? I think it's amazing. The sweetness of the pineapple with the savory pepperoni is just perfect.
Huh, I might have to give that a try sometime.
You definitely should!

Figure 12: Representative examples from the synthetic proactive-intervention dataset. Positive samples contain latent intervention opportunities that can be inferred from contextual cues, whereas negative samples are natural conversations that do not warrant assistance despite their semantic richness.

the system suppresses the proactive-action and user-state predictions for that transcript, regardless of whether the primary intervention gate has fired.

To evaluate generalization to unseen conversations, we hold out 1,000 positive and 1,000 negative transcripts from the same benchmark construction process. The positive transcripts are annotated by an LLM with gold proactive-action and user-state labels. Each transcript is embedded once as a whole, without sentence-level splitting, and scored against every state embedding using cosine similarity. For each proactive-action label, we calibrate an independent similarity threshold to recover at least 90% of its gold positives while limiting its activation rate on negative transcripts. We then measure system-level performance using the false-rejection rate, defined as the fraction of positive transcripts for which no trigger fires, and the false-acceptance rate, defined as the fraction of negative transcripts for which at least one trigger fires. We report FAR at a target FRR of 10%. Representative positive examples include a user who may miss an early train because an alarm was forgotten and a user facing uncertain weather conditions before a hike. Example conversations from the benchmark are shown in Fig. 12.

A.6 Extended Monitoring Protocol

The default PhoStream evaluation covers a relatively short interval between the query and the annotated event. Across the evaluated questions, this protocol contains 18.63 processing ticks per question on average. While appropriate for measuring response quality near the target event, it does not fully represent an always-on setting in which the system must monitor a stream for an extended period without knowing when the requested condition will occur.

We therefore construct a diagnostic extended-monitoring setting from PhoStream Forward examples. We select 25 examples from distinct videos, requiring the original query-to-gold interval to be at least $30\,\mathrm{s}$. For each example, we re-anchor the query time to the beginning of the video ($t_q = 0$), retain the original absolute gold-event time t_{gold}, and process the stream at $1\,\mathrm{s}$ intervals from the beginning of the video through the standard post-gold horizon. The selected videos have a mean duration of $946\,\mathrm{s}$, and the resulting evaluation contains 377.5 ticks per question on average.

Table 9: Effect of the decoder-gate routing criterion on PhoStream. ER, NR, and PC denote the proportions of Forward questions classified as Early Response, No Response, and Partly Correct, respectively. Qwen3.5 models are used in this experiment.

Gate	Responder	Routing criterion	Overall	Category score			Forward outcome (%)		
				Instant	Backward	Forward	ER	NR	PC
0.8B	4B	Response-contingent	30.9	63.3	59.1	0.6	98.1	0.7	1.2
		Answerability-based	30.3	55.6	46.2	9.1	38.8	44.5	16.7
2B	4B	Response-contingent	30.9	63.2	59.1	0.6	97.9	0.7	1.4
		Answerability-based	11.9	22.8	15.5	4.1	3.1	90.4	6.5
0.8B	9B	Response-contingent	35.4	68.3	65.2	4.0	89.3	3.8	6.9
		Answerability-based	33.1	60.0	50.6	10.3	38.7	44.4	16.8
2B	9B	Response-contingent	35.4	68.3	65.2	4.0	89.2	3.9	6.9
		Answerability-based	13.4	25.0	18.2	4.6	3.0	90.6	6.4

We evaluate three routing configurations. The full cascade applies the Retriever, Verifier, and Responder sequentially. The configuration without the Verifier forwards every Retriever activation directly to the Responder. The Responder-only configuration invokes the Responder at every evaluated tick. We disable sparse candidate subsampling so that all configurations operate over the same temporal sequence, and use the main pending-state threshold of $\tau^{\text{pend}} = 0.85$.

This protocol is designed to measure stage activation and inference cost rather than response accuracy. Re-anchoring extends the pre-event monitoring period while retaining the original event and response target; therefore, we use the resulting call rates only as a diagnostic of routing sparsity under extended monitoring.

A.7 Routing Criteria for Decoder-Gated Cascades

Two-model cascades offer an alternative to retrieval-based routing: a small model decides when to invoke a larger response model. LlamaPIE, for example, fine-tunes a 1B classifier to trigger an 8B model for assistance in spoken conversations (Chen et al., 2025). Its evaluations do not address multimodal intervention timing on PhoStream. We therefore test an analogous cascade using pretrained Qwen3.5-0.8B and 2B gates with Qwen3.5-4B and 9B Responders, without task-specific adaptation. At each evaluated timestep, the gate observes the available video context and user query. Under response-contingent routing, it may generate a response candidate or remain silent; the Responder is invoked only when it generates a candidate. Under answerability-based routing, the gate judges whether the evidence is sufficient to answer the query; the Responder is invoked when its probability assigned to `Yes` reaches the fixed threshold of 0.5. For each model pair, both criteria use the same checkpoints, streaming inputs, and evaluation protocol.

Response-contingent routing achieves the higher Overall score for all four model pairs and is therefore used in the main comparison. It remains prone to premature responses on Forward questions (ER: 89.2–98.1%). Answerability-based routing reduces Early Responses and improves Forward scores with the 0.8B gate ($0.6 \rightarrow 9.1$ with the 4B Responder; $4.0 \rightarrow 10.3$ with the 9B Responder), but raises Forward No Response to 44.4–44.5% and lowers Instant and Backward scores. With the 2B gate, No Response reaches 90.4–90.6%, reducing Overall performance to 11.9 and 13.4. Thus, the tested routing criteria trade premature responses for excessive deferral; we select the response-contingent criterion by the primary Overall metric while reporting the alternative's advantage on Forward timing.