PROACTIVEXBENCH: BENCHMARKING WHEN AND HOW TO INTERVENE IN THE PHYSICAL WORLD

Anonymous authors
Paper under double-blind review

ABSTRACT

Physical AI increasingly requires intelligent systems to operate autonomously in evolving real-world environments, where proactive assistance is a fundamental capability. However, existing benchmarks predominantly adopt reactive settings with explicit requests or goals, leaving proactive physical intelligence without systematic evaluation. A benchmark for this capability should assess whether a system can infer implicit intervention needs from ongoing observations, determine when intervention is appropriate, and sustain coherent communication and physical actions until the intended objective is achieved. To this end, we introduce **ProactiveXBench**, a benchmark for proactive physical intelligence comprising 1,000 real-world physical tasks with 166K fine-grained second-level annotations across diverse scenarios involving proactive communication and physical actions. ProactiveXBench evaluates the full intervention trajectory through three complementary dimensions: *Timing*, which measures alignment with the reference intervention time; *Progress*, which characterizes post-initiation execution duration; and *Outcome*, which measures event-level completion under entity-matching and ordering constraints. Evaluations of leading AI systems including GPT-6-Astra reveal substantial limitations in proactive physical intelligence: strong response or completion performance does not necessarily translate into accurate recognition of when intervention is needed, and timely initiation does not guarantee successful completion. These findings expose complementary challenges in intervention discovery and sustained execution, and establish ProactiveXBench as a unified framework for studying and advancing proactive physical intelligence.

1 INTRODUCTION

Physical AI promises intelligent systems that can perceive their surroundings, interact with people, and act in real-world environments. Realizing this promise requires more than executing explicit instructions, as systems must also recognize emerging needs and initiate appropriate assistance before being asked. Proactive physical intelligence is therefore essential for providing assistance when users may not explicitly communicate their needs. Yet systematic progress toward this capability requires an evaluation foundation that extends beyond instructed task execution. Most existing evaluations focus on responses to specified queries or goals. Recent benchmarks (Niu et al., 2025; Wang et al., 2025b;a; Zhao et al., 2026; Lu et al., 2026) have expanded evaluation to streaming understanding, response timing, and proactive interaction, but the joint assessment of autonomous intervention discovery and completion in physical environments remains underdeveloped. This gap limits our ability to systematically diagnose failures, compare systems, and guide the development of reliable proactive assistance.

A benchmark for proactive physical intelligence should assess the complete process of intervention rather than only the correctness of an isolated response. An agent must first recognize an emerging need from evolving observations under its assigned responsibilities and capabilities, then determine when to intervene and carry the response through to completion. This requires the evaluation setting to withhold explicit intervention triggers, leaving the decision of whether and when to act to the agent itself. The resulting trajectory can then be evaluated along three complementary dimensions that measure alignment with the appropriate intervention moment, capture the duration of post-initiation execution, and assess whether the required intervention is actually completed. These dimensions are evaluated separately because timely initiation does not guarantee efficient execution or successful

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

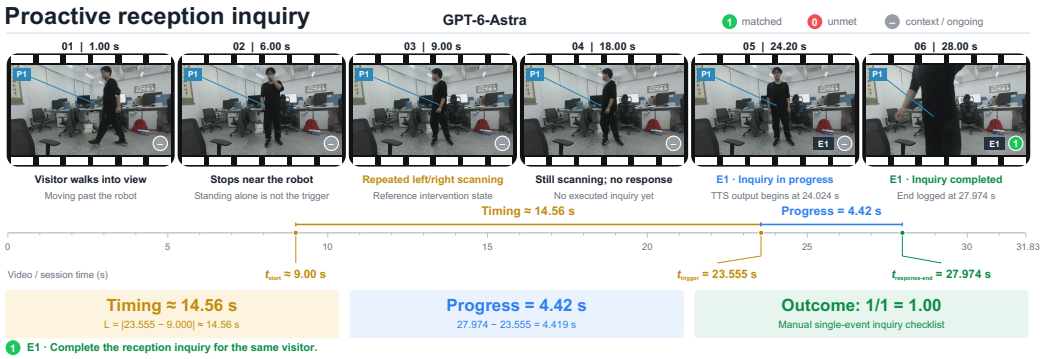


Figure 1: A real-robot reception case with GPT-6-Astra (OpenAI, 2026), illustrating delayed initiation despite successful task completion.

completion. Moreover, completion should be verified through observable, ordered events involving the intended people or objects, rather than inferred from response relevance or self-reported success.

To address this evaluation gap, we introduce **ProactiveXBench**, a benchmark that unifies recorded video observations and interactive robot-grounded scenarios under a shared proactive task formulation. It comprises 1,000 scenarios, with second-level annotations covering 166K seconds of observations. A unified construction pipeline converts evolving physical situations into tasks with implicit intervention requirements and structured event-level annotations. Agents receive roles, responsibilities, and capability constraints rather than explicit triggering conditions, and must infer intervention needs from ongoing observations. Reference intervention times and ordered, entity-associated evaluation events remain hidden during inference. Three metrics characterize the resulting intervention trajectory: *Timing* measures initiation alignment, *Progress* measures execution duration, and *Outcome* measures event completion under entity-matching and strict ordering constraints.

We conduct evaluations of leading multimodal AI systems and complement them with real-robot case studies. The results reveal weaknesses in determining when to intervene and carrying assistance through to completion. For example, in the reception case shown in Figure 1, GPT-6-Astra (OpenAI, 2026) eventually completes the required inquiry but initiates it only after a substantial delay. Claude Fable 5 (Anthropic, 2026), in contrast, fails to complete the required interventions in some interaction tasks. Across the evaluations, better initiation alignment does not consistently translate into shorter execution or more complete assistance, highlighting the importance of assessing these dimensions together. These findings point toward three priorities for future research, including recognizing implicit intervention needs, selecting context-appropriate intervention times, and sustaining execution with verifiable completion. By making these limitations measurable within a unified framework, ProactiveXBench provides a foundation for developing and evaluating more reliable proactive physical intelligence.

2 Principles and Rationales for Benchmark Construction

Proactive capability requires agents to infer intervention needs from ongoing observations and carry out appropriate assistance without an explicit request. While existing work examines proactive task prediction Lu et al. (2025), ProactiveXBench evaluates the full intervention trajectory through *Timing*, *Progress*, and *Outcome*, which characterize initiation alignment, execution duration, and ordered event completion, respectively.

2.1 Task Setting

An agent must identify and address a need for assistance in an evolving physical scenario without an explicit intervention request. At time t, it receives only the observation history $o_{1:t}$ and conditioning information c specifying its role, responsibilities, and capability constraints. Observations include visual input and audio or text when available. The agent must decide whether and when to intervene

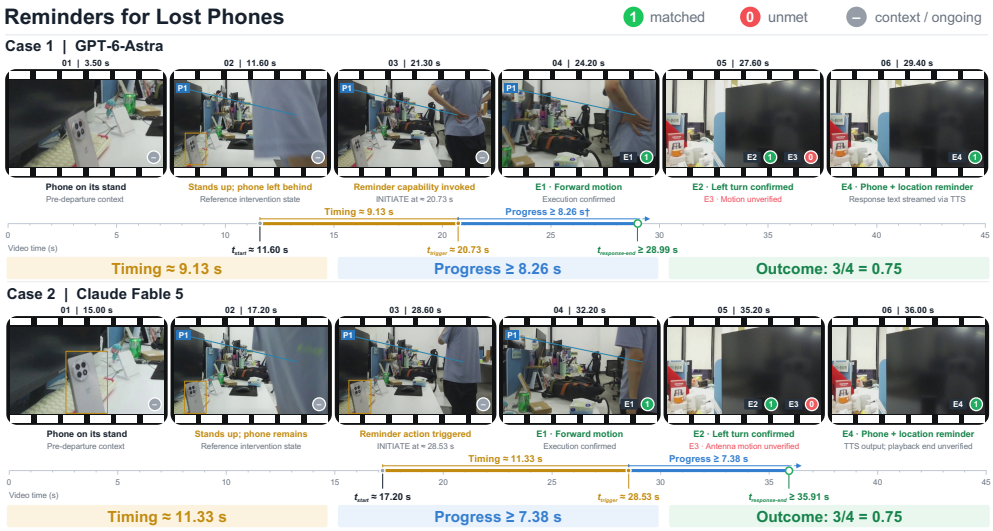


Figure 2: Real-robot cases of lost-phone reminders illustrating three complementary evaluation dimensions of proactive intervention. *Timing*, *Progress*, and *Outcome* characterize temporal alignment, execution duration, and event-level completion, respectively.

and how to complete the intervention. We characterize this process using four trajectory states:

$$a_t \in \{\text{WAIT}, \text{INITIATE}, \text{CONTINUE}, \text{STOP}\}. \tag{1}$$

These states describe continued observation, response initiation, ongoing execution, and termination without prescribing a controller or action interface. Responses may be verbal, physical, or both, subject to c. The annotated intervention need, reference intervention time, and expected response events remain hidden during inference and are used only for evaluation.

2.2 Timing: Deciding Whether and When to Intervene

Both premature and delayed interventions can undermine assistance. Reference times should therefore reflect when observable evidence justifies action under the agent's responsibilities, rather than simply when an event begins. The benchmark must preserve pre-intervention context, avoid hindsight and hidden-state assumptions, and withhold target events and triggering conditions from role descriptions. Missing responses must also remain distinct from temporal misalignment.

2.3 Progress: Executing an Intervention after Initiation

After initiation, an agent may stall, repeat information, or terminate before completing the required actions. Conversely, valid execution may require coordinating dependent steps, adapting to a person's response, or waiting. Duration alone therefore cannot distinguish efficient completion from early abandonment, or necessary interaction from an execution bottleneck. The benchmark should record initiation, termination, and intervening events without equating response termination with success. Progress characterizes execution duration, while Outcome verifies completion; interpreting them together prevents rapid but incomplete responses from being mistaken for effective assistance.

2.4 Outcome: Completing the Intended Intervention

Fluent, relevant output may still omit essential content, address the wrong participant, or violate action dependencies. Completion should therefore be grounded in observable events, their associated people or objects, and necessary ordering constraints, particularly in multi-person or multi-step interactions. These annotations specify what must be achieved without prescribing identical wording or a shared low-level action sequence. Evaluation checks whether the required events occurred, involved the appropriate entities, and respected their dependencies. When an event requires physical execution, a verbal statement of intent alone does not establish that the action occurred.

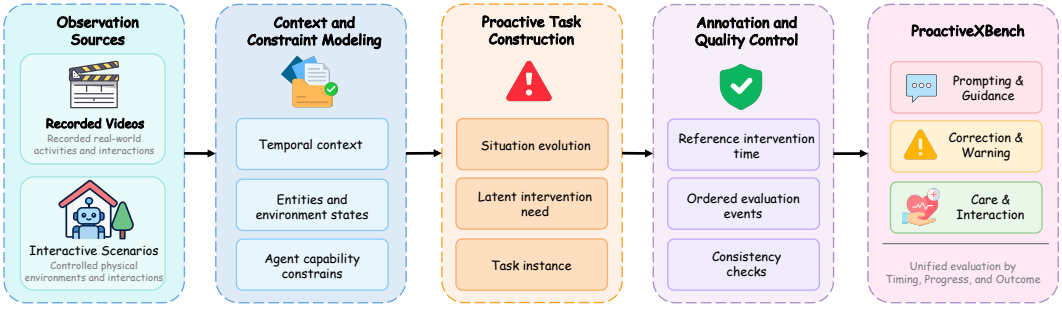


Figure 3: Unified construction pipeline of ProactiveXBench. Recorded videos and interactive scenarios undergo context and constraint modeling, model-assisted task construction, annotation, and human verification to produce proactive tasks evaluated by Timing, Progress, and Outcome.

2.5 Implications for Benchmark Design

Together, these principles preserve autonomous intervention discovery while making the resulting trajectory assessable. Agent-visible responsibilities and capabilities are separated from hidden temporal and event annotations, while response records support assessment through termination. Timing, Progress, and Outcome are defined and scored separately to distinguish temporal alignment, execution duration, and event-level completion. Strong performance on one dimension does not establish strong performance on the others. The three metrics therefore provide complementary assessments rather than interchangeable measures of intervention quality.

3 ProactiveXBench

In this section, we introduce ProactiveXBench, a unified benchmark for evaluating proactive physical intelligence across video-grounded and robot-grounded settings. Agents must infer whether and when to intervene without explicit requests, and their performance is evaluated over the full intervention trajectory rather than a single generated answer.

3.1 Proactive Task Formulation

ProactiveXBench formulates each task around the transition from an observable physical state change to a latent intervention need:

$$\text{State change} \rightarrow \text{Intervention need} \rightarrow \text{Proactive task}. \tag{2}$$

Whether an observed change warrants intervention depends on the agent's assigned responsibilities and executable capabilities. The agent must therefore infer whether and when to intervene from the evolving situation, rather than respond to an explicit request or a predefined trigger description.

Each task provides an observation sequence, an agent profile, and an interaction interface. The reference intervention time and ordered evaluation events, including their associated persons or objects, are retained as hidden annotations. This separates the information available for proactive decision making from the reference information used to evaluate the resulting behavior. Appendix A.1 details the task components and information visibility.

3.2 Task Construction Pipeline

As illustrated in Figure 3, recorded videos and interactive scenarios are processed through five shared stages: observation collection, context and constraint modeling, proactive task construction, annotation, and quality control. Candidate situations are represented through their temporal context, relevant entities, environmental states, and feasible actions, then reformulated as role-conditioned proactive tasks.

Task construction and annotation are model-assisted, with all proposals subsequently reviewed and revised by human annotators. Human verification ensures that each retained task has an observable

intervention need and a response compatible with the agent's responsibilities and capabilities. Verified tasks are organized into *Guidance*, *Warning*, and *Interaction*, and evaluated consistently using **Timing**, **Progress**, and **Outcome**. Observation sources, construction procedures, and verification criteria are detailed in Appendix A.2.

3.3 Annotation and Quality Control

Each task is annotated with a reference intervention time and an ordered sequence of evaluation events. The reference time marks when observable evidence is sufficient to warrant intervention, while the events specify the expected response content, associated persons or objects, and required ordering or prerequisite relations. These annotations remain hidden from the agent during inference.

Quality control checks source and temporal validity, evidence observability, capability compatibility, and the presence of a meaningful intervention need. Agent-facing descriptions are also reviewed to prevent disclosure of task-specific triggers or reference annotations. Data splits are kept disjoint at the source- or scenario-group level. Detailed annotation are provided in Appendix A.3.

3.4 Metric Design

Proactive interventions in digital and physical environments share a common temporal and goal-oriented structure: agents must infer intervention needs from evolving context, determine when to respond, and execute appropriate responses to achieve task goals. This shared structure provides a modality-agnostic basis for evaluation through response timestamps and task-relevant execution evidence. Accordingly, we define three unified dimensions: **Timing** measures the absolute deviation from the reference response time, **Progress** measures intervention execution time, and **Outcome** assesses task completion through event matching under execution-order constraints.

Timing. Timing measures the absolute response latency:

$$L = \left| t_{\text{trigger}} - t_{\text{start}} \right|, \tag{3}$$

where t_{start} denotes the reference time at which an intervention is needed, and t_{trigger} denotes the time at which the agent initiates its proactive response. Early and late responses are penalized symmetrically according to their absolute temporal deviation, with smaller values indicating better temporal alignment.

Progress. Progress measures the time required to execute an intervention after initiation:

$$T_{\text{exec}} = t_{\text{response-end}} - t_{\text{trigger}}, \tag{4}$$

where $t_{\text{response-end}}$ denotes the time at which the language response or physical action required for the intervention is completed. This metric captures the execution duration from the initiation of a proactive response to its completion.

Outcome. Outcome measures task completion based on event matching under execution-order constraints:

$$S_{\text{Outcome}} = \begin{cases} \dfrac{N_{\text{match}}}{N_{\text{total}}}, & \text{if the order is valid,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Here, N_{match} denotes the number of events for which both the event content and the associated person match the reference annotations. N_{total} denotes the total number of predefined events to be evaluated, including events that are not executed. When the prescribed execution order is followed, the score is the proportion of matched events among all predefined events. Any violation of the execution-order constraints results in a score of zero. The score ranges from 0 to 1, with higher values indicating greater task completion.

3.5 Benchmark Statistics

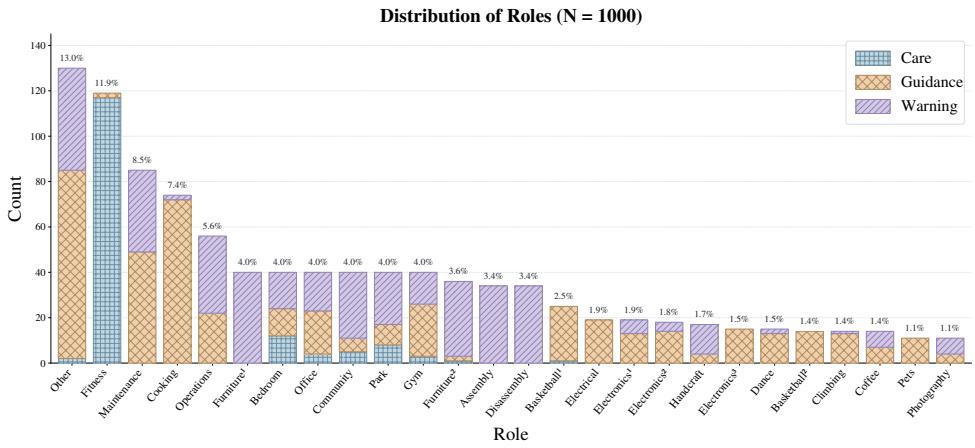


Figure 4: Overall task distribution in ProactiveXBench. The inner ring summarizes the three functional categories across 1,000 tasks, while the outer ring shows their finer-grained role assignments.

Domain	Tasks number	Avg. steps per task	video duration (s)
Guidance	438	11.83	71,999
Warning	409	14.64	74,920
Interaction	153	23.37	19,171
Total	**1000**	**14.62**	**166,090**

Table 1: Statistics of tasks across three domains.

Figure 4 summarizes the task distribution in ProactiveXBench across three functional categories and their finer-grained role assignments. *Guidance* covers tasks requiring timely instructions or assistance to support ongoing activities; *Warning* focuses on proactive alerts and corrective interventions in response to potential risks or errors; and *Care* involves recognizing individual needs and initiating supportive or social interactions. Detailed statistics are provided in Appendix A.4.

4 Experiemental

4.1 Experimental Setup

We evaluate all systems under the same proactive interaction protocol. During inference, each model receives only the observable environment context, its assigned responsibilities, and the corresponding capability constraints. Reference intervention times, target events, and completion annotations are hidden. The model continuously processes the evolving observation and determines whether to WAIT, INITIATE, CONTINUE, or STOP.

Compared systems. We compare Proactive-Engine, a specialized proactive system, with six representative multimodal baselines: Omni-3B(Xu et al., 2025), Qwen3.5-4B (Qwen Team, 2026b), Qwen3.5-27B (Qwen Team, 2026a), GPT-6-Astra (OpenAI, 2026), Gemini 3 (Pichai et al., 2025), and Claude Fable 5 (Anthropic, 2026). Proactive-Engine adopts a controller–worker architecture that separates intervention timing from respon se generation: the controller determines when to initiate or terminate an intervention, while the worker generates verbal responses or invokes available physical actions. All systems receive the same observable information and are evaluated without access to reference intervention times or annotated evaluation events. Model descriptions, selection rationale, and methodological details are provided in Appendix B.

Robotic platform. Physical execution is carried out on a Reachy Mini-based platform with a predefined capability profile. Its action space includes verbal communication, person-oriented motion, approaching, pointing, stopping, and interaction termination, enabling us to evaluate proactive decision making and sustained intervention without introducing the additional complexity of low-level manipulation or locomotion. Physical actions are credited only when verified from the execution trajectory; textual claims or stated intentions alone do not count as successful execution.

Evaluation protocol. Performance is evaluated using three complementary metrics: **Timing**, **Progress**, and **Outcome**. Timing measures the alignment of intervention initiation with the ref-

Domain	Model	Timing				Progress			Outcome ↑
		Mean ↓	Median ↓	P80 ↓	≤ 2 s ↑	Mean ↓	Median ↓	P80 ↓	
Guidance	GPT-6-Astra	11.98	10.16	14.90	3.2%	3.83	3.82	4.75	**0.97**
	Gemini 3	4.92	3.97	6.60	12.7%	4.22	4.03	5.11	0.96
	Claude Fable 5	10.92	9.48	13.66	1.6%	4.42	4.43	5.36	0.94
	Qwen3.5VL 27B	12.46	11.13	16.92	0.0%	3.74	3.47	4.55	0.92
	ProPAI-Engine	**1.33**	**1.29**	**1.66**	**92.1%**	**3.35**	**3.19**	**3.96**	0.93
Warning	GPT-6-Astra	4.28	3.85	6.06	14.0%	5.90	5.69	6.98	**0.96**
	Gemini 3	1.83	1.41	2.53	63.6%	5.98	5.62	7.47	0.95
	Claude Fable 5	16.52	13.22	24.49	0.0%	6.28	5.85	8.29	0.92
	Qwen3.5VL 27B	2.36	2.09	3.06	48.6%	5.81	5.43	7.30	0.93
	ProPAI-Engine	**1.22**	**1.07**	**1.56**	**91.6%**	**3.53**	**3.25**	**4.31**	0.94
Interaction	GPT-6-Astra	12.15	10.63	15.48	0.0%	11.89	10.43	14.55	**0.95**
	Gemini 3	7.34	6.90	10.03	0.0%	13.27	13.09	16.15	0.94
	Claude Fable 5	7.50	5.33	10.65	6.7%	12.26	10.71	14.98	0.93
	Qwen3.5VL 27B	7.37	6.71	10.61	0.0%	12.13	11.40	15.00	0.91
	ProPAI-Engine	**1.73**	**1.53**	**2.14**	**66.7%**	**8.36**	**7.76**	**10.84**	0.89

Table 2: Illustrative domain-wise results for real-robot scenarios on ProactiveXBench using synthetic data. Timing and Progress are reported in seconds, P80 uses linear interpolation, and ≤ 2 s denotes the percentage of timely interventions. Outcome values are illustrative synthetic scores. Best values within each domain are shown in bold.

erence intervention point, Progress captures the duration required to complete the intervention, and Outcome evaluates event-level task completion under ordering and entity-matching constraints. Temporal metrics are reported in seconds, while missing measurements are reported separately from zero scores.

4.2 Main Results

We evaluate proactive intervention in terms of initiation timing, execution duration, and event-level completion. Results for real-robot testing and physical-scenario evaluation are reported separately in Tables 2 and 3, respectively, to account for differences in task distributions and execution conditions. Timing and Progress are interpreted alongside the corresponding eligible sample counts and Outcome scores, since prompt initiation or short execution alone does not establish successful intervention.

Real-robot testing. The reported results reveal gaps in both intervention timing and event-level completion across the three domains, as summarized in Table 2. GPT-6-Astra records mean Timing errors of 11.98 s, 4.28 s, and 12.15 s in *Guidance*, *Warning*, and *Interaction*, respectively, with corresponding Outcome scores of 0.78, 0.74, and 0.64. These scores indicate that its interventions deviate from the reference initiation time and do not fully satisfy the event-level requirements. The limitations are most pronounced in *Interaction*, where its largest mean Timing error coincides with its lowest Outcome score.

Execution duration provides a complementary but insufficient measure of intervention quality. In *Guidance*, GPT-6-Astra records a Progress median of 3.82 s, yet this brief post-initiation duration coexists with substantial timing error and incomplete event fulfillment. A short intervention therefore should not, by itself, be interpreted as effective assistance. Proactive-Engine achieves lower mean Timing errors and higher Outcome scores across all three domains, reaching Outcome scores of 0.93, 0.94, and 0.89, respectively. Its relative advantage nevertheless leaves room for improvement, since not all intervention requirements are satisfied. These patterns suggest that reliably coordinating appropriately timed initiation with complete execution remains a challenge for the evaluated systems, including GPT-6-Astra, rather than one resolved by shorter execution alone.

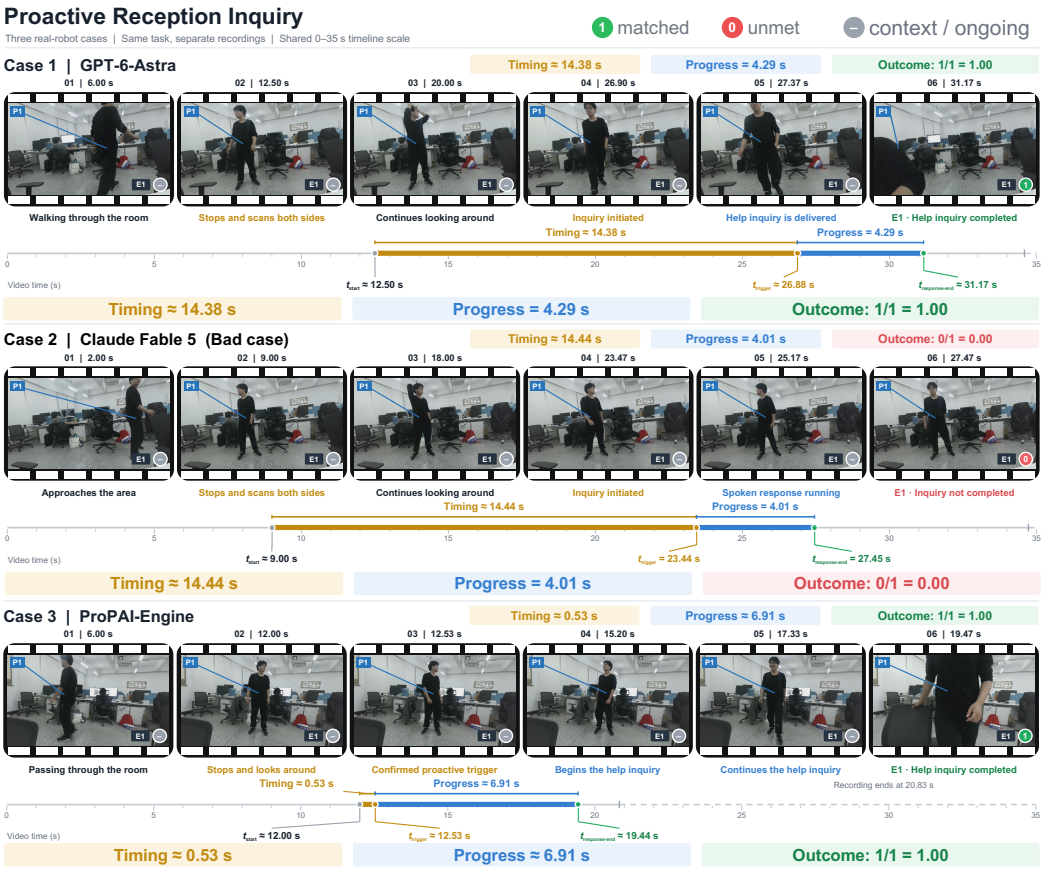


Figure 5: Cases of proactive reception inquiry, annotated with Timing, Progress, and Outcome.

Physical-scenario evaluation. Table 3 shows that Proactive-Engine consistently achieves lower Progress medians and higher Outcome scores across all three domains, while its Timing advantage varies. For example, in *Warning*, it achieves a Progress median of 2.10 s and an Outcome score of 0.193, compared with 44.10 s and 0.057 for Qwen3.5-4B, despite the latter's lower mean Timing error. A similar contrast appears in *Interaction*, where Omni-3B achieves lower Timing error but exhibits substantially longer execution durations and poorer event completion.

These results show that accurate initiation timing alone does not ensure successful assistance, and shorter execution durations should be assessed alongside event completion. Despite Proactive-Engine's relative gains, the low Outcome scores across all systems underscore the difficulty of completing multi-step proactive interventions.

4.3 Implicit Intervention Formulation

We further examine the role of implicit intervention formulation by comparing identity-driven prompts with paired explicit-condition variants. Both settings share the same observations, reference intervention time, and evaluation events, while differing only in whether the task-specific intervention condition is revealed. The explicit formulation turns intervention into recognizing a predefined condition, whereas the identity-driven formulation requires the agent to determine from its role and the evolving context whether intervention is warranted. The comparison therefore separates intervention-need inference from simple trigger recognition. By withholding the task-specific condition, the identity-driven setting places greater emphasis on autonomous intervention discovery, which is the capability ProactiveXBench is designed to evaluate.

Domain	Model	Timing				Progress			Outcome ↑
		Mean ↓	Median ↓	P80 ↓	≤2s ↑	Mean ↓	Median ↓	P80 ↓	
Guidance	ProPAI-Engine	**3.19**	1.10	**3.10**	**63.6%**	**21.58**	**4.00**	**23.97**	**0.309**
	Omni-3B	7.76	**1.00**	20.00	54.5%	80.60	34.50	144.00	0.011
	Qwen3.5-4B	5.30	2.10	6.00	46.9%	97.31	65.00	185.60	0.052
Warning	ProPAI-Engine	1.72	1.10	**3.10**	55.7%	**2.79**	**2.10**	**4.10**	**0.193**
	Omni-3B	1.95	**0.11**	4.10	52.2%	144.65	104.10	279.10	0.019
	Qwen3.5-4B	**1.43**	1.10	**3.10**	**63.0%**	92.42	44.10	158.10	0.057
Interaction	ProPAI-Engine	3.08	2.50	4.50	43.0%	**16.04**	**13.00**	**23.50**	**0.087**
	Omni-3B	**1.73**	**1.50**	**2.50**	**72.7%**	99.09	111.00	146.50	0.004
	Qwen3.5-4B	2.55	2.00	4.50	56.7%	61.08	50.50	117.50	0.021

Table 3: Domain-wise results on the physical-scenario evaluation of ProactiveXBench. Timing and Progress are reported in seconds, while ≤2s denotes the proportion of interventions initiated within two seconds of the reference time. Best results within each domain are shown in bold.

4.4 Process-Level Analysis

Initiation alignment and intervention completion reveal distinct limitations. In *Interaction*, Omni-3B achieves the lowest mean Timing error in Table 3, yet its Outcome is only 0.004. Conversely, GPT-6-Astra completes the reception inquiry despite initiating over 14 s late. Neither timely initiation nor eventual completion alone establishes effective proactive assistance.

Similar execution durations can conceal different completion states. In the reception cases, GPT-6-Astra and Claude Fable 5 record Progress durations of 4.29 s and 4.01 s, but Outcome scores of 1.00 and 0.00, respectively. The slightly shorter response does not indicate greater efficiency, since the required inquiry remains incomplete. Progress must therefore be interpreted alongside Outcome to distinguish brief execution from successful completion.

Event-level evidence identifies unsupported requirements. In both phone-reminder cases, the verbal reminder and two motion events are credited, but another motion event remains unverified, yielding an Outcome of 0.75. Relevant reminders therefore do not establish full intervention completion. Timing, Progress, and Outcome separately assess initiation alignment, post-initiation duration, and ordered, entity-matched event completion.

4.5 Real-Robot Case Studies

We conduct real-robot case studies across three scenarios to examine proactive intervention from initiation to completion. Detailed trajectory analyses are provided in Appendix B.3.

5 Conclusion

We introduced ProactiveXBench, a benchmark for evaluating proactive physical intelligence through the autonomous discovery and completion of context-dependent interventions. Across 1,000 tasks spanning three functional domains, agents must infer intervention needs from evolving observations and act within assigned responsibilities and capability constraints, without explicit triggers. A unified construction pipeline and hidden reference annotations support trajectory-level evaluation through Timing, Progress, and Outcome. Experiments reveal that better initiation alignment does not necessarily translate into shorter execution or higher event-level completion, with real-robot testing further illustrating the gap between initiating assistance and completing an intervention. These findings highlight the importance of evaluating proactive assistance beyond its initial response and provide a foundation for developing agents that recognize emerging needs and sustain appropriate interventions through completion.

AI USE STATEMENT

Generative AI tools were used to aid or polish writing.

REFERENCES

Anthropic. Claude Fable 5 and Claude Mythos 5. Official model announcement, June 2026. URL https://www.anthropic.com/news/claude-fable-5-mythos-5. Published June 9, 2026. Accessed September 21, 2026.

Kuofei Fang, Xinyi Che, Haomin Ouyang, Shufan Zhang, Xuehao Wang, Qi Liu, Liyi Liu, Chenqi Zhang, Wenxi Cai, Wenyu Dai, et al. Roboteq: Transitioning from passive intelligence to active intelligence in embodied ai. *arXiv preprint arXiv:2605.06234*, 2026.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zachary Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, María Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Dutt 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 González, 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 K. 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 Arbeláez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Julian Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard A. Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shout, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. *Int. J. Comput. Vis.*, 133(12):8356–8435, 2025. doi: 10.1007/S11263-025-02557-6. URL https://doi.org/10.1007/s11263-025-02557-6.

Raphael Lorenzo-Louis, Fabio Amadio, Bertrand Luvison, and Serena Ivaldi. Hui360: A 360° egocentric dataset and baselines for human-robot interaction anticipation. In *2026 IEEE 20th International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–9. IEEE, 2026.

Xudong Lu, Xueying Li, Annan Wang, Yang Bo, Jinpeng Chen, Zengliang Li, Nianzu Yang, Rui Liu, Xue Yang, Jingwen Hou, and Hongsheng Li. Omniinteract: Benchmarking real-world streaming interaction for real-time omnimodal assistants. *CoRR*, abs/2605.26485, 2026. doi: 10.48550/ARXIV.2605.26485. URL https://doi.org/10.48550/arXiv.2605.26485.

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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=sRIU6k2TcU.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. Ovo-bench: How far is your video-llms from real-world online video understanding? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 18902–18913. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01761. URL https://openaccess.thecvf.com/content/CVPR2025/html/Niu_OVO-Bench_How_Far_is_Your_Video-LLMs_from_Real-World_Online_Video_CVPR_2025_paper.html.

OpenAI. GPT-6 Astra System Card. System card, OpenAI, September 2026. URL https://deploymentsafety.openai.com/gpt-6-astra/gpt-6-astra.pdf. Published September 3, 2026. Accessed September 21, 2026.

Sunny Panchal, Apratim Bhattacharyya, Guillaume Berger, Antoine Mercier, Cornelius Böhm, Florian Dietrichkeit, Reza Pourreza, Xuanlin Li, Pulkit Madan, Mingu Lee, Mark Todorovich, Ingo Bax, and Roland Memisevic. What to say and when to say it: Live fitness coaching as a testbed for situated interaction. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/8aa695b5d6bf9e958d4e56b22c9b859d-Abstract-Datasets_and_Benchmarks_Track.html.

Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with Gemini 3. Google Blog, November 2025. URL https://blog.google/products-and-platforms/products/gemini/gemini-3/. Published November 18, 2025. Accessed September 21, 2026.

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

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

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 1207–1216. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00130. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Tang_COIN_A_Large-Scale_Dataset_for_Comprehensive_Instructional_Video_Analysis_CVPR_2019_paper.html.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. Holoassist: an egocentric human interaction dataset for interactive AI assistants in the real world. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 20213–20224. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01854. URL https://doi.org/10.1109/ICCV51070.2023.01854.

Yueqian Wang, Xiaojun Meng, Yifan Wang, Huishuai Zhang, and Dongyan Zhao. Proactivevideoqa: A comprehensive benchmark evaluating proactive interactions in video large language models. *CoRR*, abs/2507.09313, 2025a. doi: 10.48550/ARXIV.2507.09313. URL https://doi.org/10.48550/arXiv.2507.09313.

Yuxuan Wang, Yueqian Wang, Bo Chen, Tong Wu, Dongyan Zhao, and Zilong Zheng. Omnimmi: A comprehensive multi-modal interaction benchmark in streaming video contexts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 18925–18935. Computer Vision Foundation / IEEE, 2025b. doi: 10.1109/CVPR52734.2025.01763. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_OmniMMI_A_Comprehensive_Multi-modal_Interaction_Benchmark_in_Streaming_Video_Contexts_CVPR_2025_paper.html.

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. *CoRR*, abs/2503.20215, 2025. doi: 10.48550/ARXIV.2503.20215. URL https://doi.org/10.48550/arXiv.2503.20215.

Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. HACS: human action clips and segments dataset for recognition and temporal localization. In *2019 IEEE/CVF International*

Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 8667–8677. IEEE, 2019. doi: 10.1109/ICCV.2019.00876. URL https://doi.org/10.1109/ICCV.2019.00876.

Ruixiang Zhao, Jie Yang, Zijie Xin, Tianyi Wang, Fengyun Rao, Jing Lyu, and Xirong Li. Omnipro: A comprehensive benchmark for omni-proactive streaming video understanding. *CoRR*, abs/2605.18577, 2026. doi: 10.48550/ARXIV.2605.18577. URL https://doi.org/10.48550/arXiv.2605.18577.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 7590–7598. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.12342. URL https://doi.org/10.1609/aaai.v32i1.12342.

A Task Formulation and Construction Details

A.1 Task Representation

Role-dependent intervention needs. Candidate intervention opportunities arise from observable changes, such as a person entering a potentially unsafe state, an object being left behind, a visitor showing signs of needing assistance, or an ongoing activity reaching a stage where guidance becomes appropriate. These observations do not constitute fixed intervention triggers. Their relevance depends on the agent's assigned role, responsibilities, and executable capabilities. The task therefore requires interpreting the observed situation in relation to the agent's responsibilities, rather than matching it to an explicitly stated trigger.

Instance components. Each task instance is organized into five components. The *observation information* records the video or scenario sequence. The *agent conditioning* specifies the assigned identity, responsibilities, and executable capabilities. The *intervention annotation* records the latent intervention need and its reference time. The *evaluation events* define the expected actions or responses, their associated persons or objects, and their required temporal order. The *provenance information* preserves source and scenario relationships for constructing disjoint evaluation partitions.

Information visibility. The observation sequence and agent conditioning are provided to the evaluated model together with an interaction interface specifying its available communication or action options. Agent-facing descriptions define general responsibilities without naming the task-specific trigger condition or prescribing the reference response events. Intervention and event annotations are reserved for evaluation and remain hidden throughout inference.

A.2 Task Construction

Observation sources. The recorded-video portion contains 800 video-level task instances derived from six human-activity datasets: COIN (Tang et al., 2019), Ego-Exo4D (Grauman et al., 2025), HACS (Zhao et al., 2019), HoloAssist (Wang et al., 2023), YouCook2 (Zhou et al., 2018), and QEVD-FIT-COACH (Panchal et al., 2024). These datasets provide procedural, assistive, and interactive activities with diverse temporal structure. Their original task objectives are not retained; instead, the observations are reinterpreted as evolving physical contexts in which intervention needs may emerge. Interactive scenarios complement these recordings with controlled settings in which human behaviors, object states, environmental changes, and agent interactions can be systematically instantiated while remaining directly observable to the agent.

Context and constraint modeling. For each candidate situation, context modeling captures the temporal evolution of relevant entities and environmental states. Capability constraints delimit the set of feasible interventions, ensuring that the task does not require responses beyond the agent's available actions. Task construction then specifies the agent's identity, responsibilities, and admissible behaviors while leaving the intervention opportunity implicit in the observation sequence.

Model-assisted construction and annotation. Multimodal model assistance is used to produce candidate task descriptions and annotation proposals. These outputs serve as intermediate proposals rather than finalized benchmark instances. Human annotators review and revise the proposals, finalizing the reference intervention time, expected response events, associated persons or objects, and required temporal order.

Human verification. Each candidate is checked for observability, temporal consistency, intervention necessity, role appropriateness, and compatibility with the agent's executable capabilities. Observability requires the intervention need to be supported by evidence available in the observation sequence rather than hidden state. Temporal consistency requires reference times and event ordering to agree with the observed situation. Intervention necessity and role appropriateness distinguish meaningful assistance from ordinary activity that does not warrant action by the assigned agent. Capability compatibility ensures that the expected response can be carried out through the available interaction interface. Only verified instances are retained in the benchmark.

Functional organization. Verified tasks are grouped by intervention function rather than observation source. *Prompting & Guidance* covers timely information, instruction, and assistance. *Correction & Warning* addresses emerging errors and risks. *Care & Interaction* concerns human needs and evolving interaction states. This shared organization supports consistent evaluation across different observation settings.

A.3 Annotation Protocol and Quality-Control Details

Temporal grounding. For recorded observations, source temporal boundaries provide initial anchors for the reference intervention time. These anchors are checked against the visible evidence and, where necessary, aligned to the available observation sequence. For interactive scenarios, the reference time is grounded in the observable state transition that establishes the intervention need. The evaluated agent's response initiation and completion times are extracted from its generated trajectory rather than taken from the reference annotations.

Source and temporal validation. The first quality-control stage removes instances with missing or undecodable media, invalid or out-of-range timestamps, or inconsistent scenario transitions. This stage ensures that each retained instance has a valid observation sequence and a coherent temporal reference.

Observability review. The second stage verifies that the evidence needed to identify the intervention opportunity is accessible through the supplied observations. It also checks that the expected response is compatible with the agent's assigned capabilities, excluding tasks that depend on hidden information or unavailable actions.

Intervention-need review. The third stage distinguishes meaningful intervention opportunities from ordinary action occurrences. Cases are excluded when they merely depict an activity without establishing a need for proactive assistance under the agent's assigned role and responsibilities.

Prompt review. The fourth stage checks that agent-facing descriptions provide general roles, responsibilities, and capability constraints without revealing the task-specific intervention trigger, reference timing, or expected response. This separation preserves the requirement that the agent infer the intervention need from the evolving observations.

Supplementary review and provenance. A supplementary model-assisted review checks role consistency, intervention relevance, and potential leakage of reference information. This review serves as an internal consistency check rather than an independent estimate of annotation reliability. Provenance information is retained for every accepted instance to support source- or scenario-group separation across data splits.

A.4 Detailed Role Distribution

Frequently represented roles in ProactiveXBench include fitness coach, equipment maintenance guide, cooking instructor, and equipment operations guide, while many specialized responsibilities occur at lower frequencies. The benchmark also includes five environment-specific assistant roles, each contributing 40 task instances and extending coverage to everyday physical settings. Together, these roles span instruction, coaching, equipment assistance, and situational interaction. Their unequal frequencies reflect the benchmark's non-uniform role coverage rather than a balanced allocation across all responsibilities.

B Compared Systems and Selection Rationale

B.1 General-Purpose Multimodal Baselines

We select six general-purpose multimodal baselines spanning different model scales and families to examine whether visual understanding, instruction following, and multimodal reasoning transfer to proactive assistance. Their inclusion does not assume that general multimodal competence guarantees appropriate intervention timing or successful task completion.

Omni-3B. Omni-3B (Xu et al., 2025) supports image, audio, and video understanding alongside response generation. It serves as a compact multimodal baseline for assessing whether observation understanding supports the recognition of implicit intervention needs.

Qwen3.5-4B. Qwen3.5-4B (Qwen Team, 2026b) combines visual understanding with instruction following. We include it to examine whether a compact vision-language model can infer intervention needs under role and capability constraints.

Qwen3.5-27B. Qwen3.5-27B (Qwen Team, 2026a) provides a larger open-weight vision-language baseline. Together with Qwen3.5-4B, it enables comparison of proactive performance across model scales within the same family.

GPT-6-Astra. GPT-6-Astra (OpenAI, 2026) combines image understanding with complex reasoning. It is included to assess whether these capabilities support implicit need recognition and multistep assistance without an explicit user request.

Gemini 3. Gemini 3 (Pichai et al., 2025) supports video understanding and multimodal reasoning. We include it to examine whether interpreting temporal context helps identify intervention opportunities in evolving situations.

Claude Fable 5. Claude Fable 5 (Anthropic, 2026) supports visual understanding and long-horizon agentic reasoning. It serves as a baseline for assessing whether these capabilities translate into coherent assistance across successive observations and interaction feedback.

B.2 Proactive-Engine

Timing and response coordination. Proactive-Engine serves as a specialized proactive reference that explicitly coordinates intervention timing with response generation. Its controller–worker architecture comprises a Proactive Timing Model and a Proactive Response Model. The Timing Model is post-trained on video-derived causal decision sequences to predict `<silence>` or `<response>` from the observation history, task or role context, and current output state. Depending on whether a response is active, these decisions correspond to waiting, initiation, continuation, or output release. The Response Model asynchronously processes the same observation stream and incrementally caches context, allowing a timing trigger to initiate response generation without recomputing the full history. Observation processing continues during generation and execution.

Capability-grounded response and feedback. The Response Model adapts a pretrained multimodal backbone through contextual prompting rather than separate response-model training. It

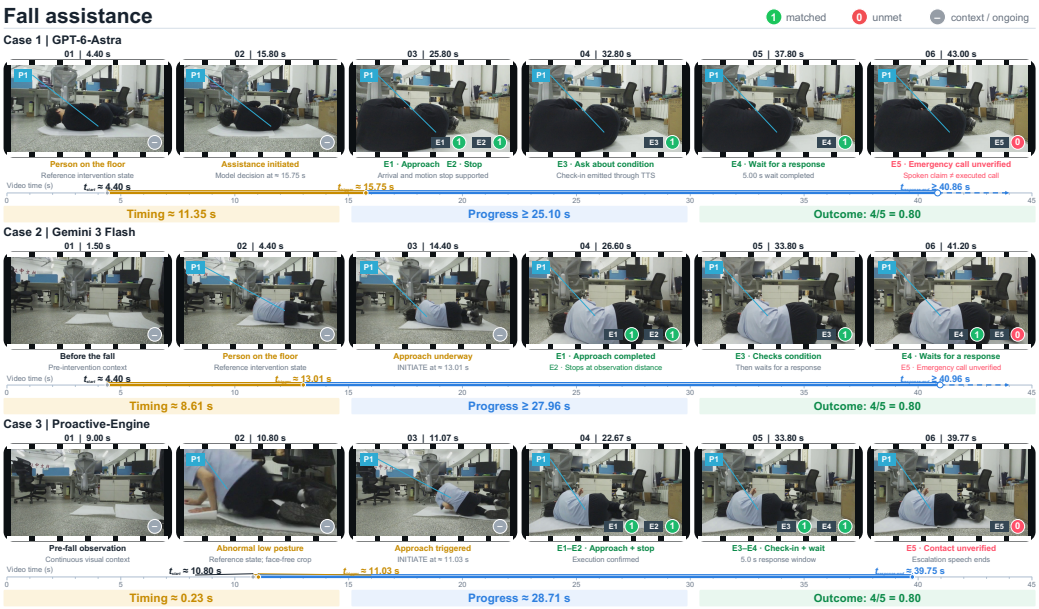


Figure 6: Real-robot fall-assistance trajectories evaluated with Timing, Progress, and Outcome.

generates human-facing text, converted to speech by an external text-to-speech module, or invokes available device capabilities through typed functions in the Physical Coding Interface (PCI). A persistent runtime maintains task states and execution histories, validates and dispatches function calls, and incorporates subsequent observations and execution results into the response context. This feedback supports follow-up actions, response revision, and intervention closure, connecting continuous observation with capability-grounded assistance.

B.3 Real-Robot Case Studies

We analyze three real-robot scenarios to examine how intervention timing, response execution, and event completion interact in practice: fall assistance, forgotten-belongings reminders, and proactive reception inquiries. In each scenario, the robot must infer an intervention need from observable context without an explicit request. The annotated trajectories distinguish initiating an appropriate response from completing the events required by the task. All timestamps are relative to the corresponding recording, and Progress is reported as a lower bound when the response endpoint cannot be verified.

B.3.1 Fall Assistance

Scenario. A person falls without explicitly requesting help, requiring the robot to recognize the need for assistance from the observed situation, as illustrated in Figure 6. The annotated intervention comprises approaching the person, stopping at an appropriate distance, checking their condition, waiting for a response, and contacting emergency assistance.

Trajectory and evaluation. Proactive-Engine initiates close to the reference time, whereas GPT-6-Astra and Gemini 3 Flash respond substantially later. Despite this difference, all three trajectories receive credit for the first four events, while emergency contact remains unverified. Each therefore obtains an Outcome score of 0.80.

Analysis. The trajectories differ in when assistance begins, but not in the extent of verified completion. Prompt initiation allows assistance to begin earlier without resolving the remaining gap in the required sequence. In particular, a spoken claim of emergency contact cannot substitute for evidence that the action occurred. This case thus distinguishes responsiveness from completion grounded in physical execution.

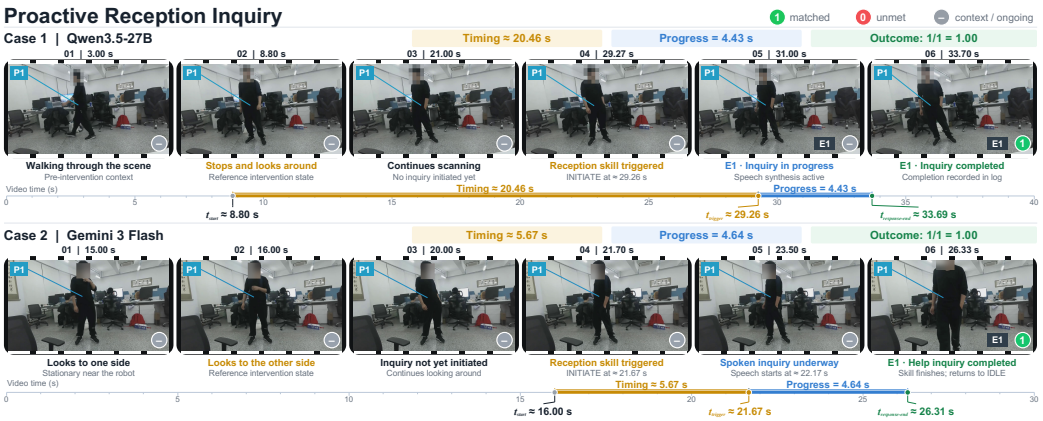


Figure 7: Real-robot forgotten-phone reminder trajectories evaluated with Timing, Progress, and Outcome.

B.3.2 Forgotten-Belongings Reminder

Scenario and intervention requirements. A person prepares to leave while a phone remains behind, creating an implicit need for intervention, as shown in Figure 7. The robot must relate the person's departure to the unattended object and infer the need for a reminder. The prescribed response combines forward motion, a left turn, and antenna movement with a spoken reminder identifying the phone and its location.

Trajectory and evaluation. Qwen3.5-27B and Gemini 3 Flash Preview both perform the forward motion and turn and issue the phone-and-location reminder. Antenna movement remains unverified, giving both systems an Outcome score of 0.75. Qwen3.5-27B initiates closer to its reference time, but this difference does not change the credited event coverage.

Analysis. These responses connect the departing person with the relevant object, yet do not establish completion of the full embodied sequence. The case highlights the distinction between producing contextually appropriate language and satisfying the accompanying action requirements. The partial score concerns this prescribed reminder sequence, not whether the phone is subsequently retrieved, which lies outside the annotated objective.

B.3.3 Proactive Reception Inquiry

Scenario and intervention requirements. A person pauses near the robot and looks around without directly requesting assistance, as illustrated in Figure 8. The robot must recognize a possible service need and offer help to the relevant person. The annotated objective is completion of the spoken inquiry, rather than a subsequent guidance interaction.

Trajectory and evaluation. Qwen3.5-27B and Gemini 3 Flash both complete the inquiry and obtain an Outcome score of 1.00. Their response durations are similar, but Gemini 3 Flash initiates substantially closer to its reference time. Qwen3.5-27B leaves a longer interval between the observable intervention need and its offer of assistance.

Analysis. The principal difference lies before initiation rather than during the spoken response. Identical completion scores and similar execution durations therefore conceal a substantial difference in responsiveness. This case complements fall assistance: whereas timely initiation need not lead to full completion, completing the required response need not imply that assistance was offered promptly.

Discussion. Together, these cases distinguish timely but partially verified intervention from completed but delayed assistance. They also show why plausible language and sustained activity are

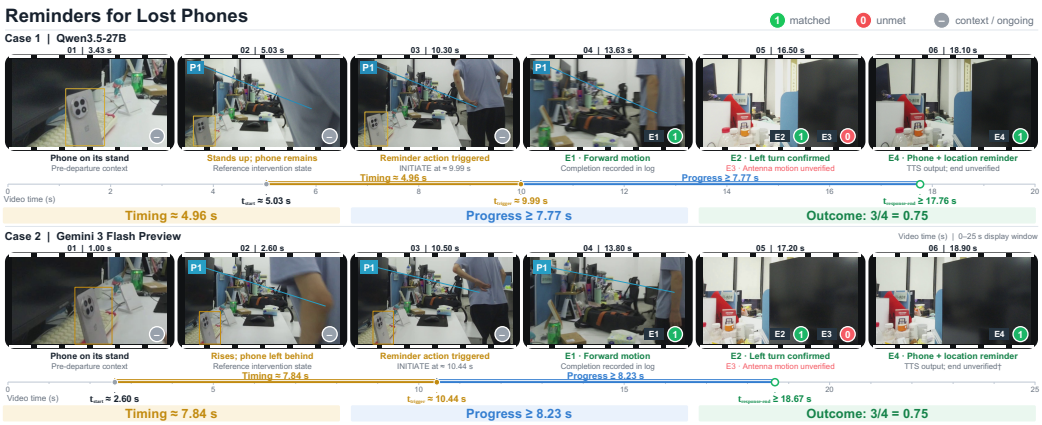


Figure 8: Real-robot proactive reception-inquiry trajectories evaluated with Timing, Progress, and Outcome.

insufficient evidence of task completion. Joint evaluation through Timing, Progress, and Outcome makes these differences explicit while keeping conclusions grounded in the annotated objective and available execution evidence.

C Related Work

Online understanding and proactive interaction. Existing benchmarks evaluate online understanding, autonomous response timing, and interaction continuity in streaming settings. OVO-Bench (Niu et al., 2025) examines temporal reasoning in online video understanding, while OmniMMI (Wang et al., 2025b) evaluates streaming multimodal understanding and proactive reasoning. OmniPro (Zhao et al., 2026) focuses on autonomous response decisions, and ProactiveVideoQA (Wang et al., 2025a) assesses time sensitive, multi-turn interactions. OmniInteract (Lu et al., 2026) further evaluates response correctness, timing, interruption handling, and continuity across ongoing interactions. Together, these benchmarks examine how systems understand evolving inputs and coordinate responses over time.

Intention anticipation and behavior judgment. Complementary benchmarks examine the human intentions and contextual constraints that inform proactive behavior. HUI360 (Lorenzo-Louis et al., 2026) evaluates human–robot interaction intention anticipation from egocentric observations. RobotEQ (Fang et al., 2026) assesses the social appropriateness of robot actions through behavior judgment and spatial grounding. These benchmarks evaluate important foundations for deciding whether and how to assist, while anticipating an interaction or identifying an appropriate action does not by itself establish successful assistance.

Building on these complementary directions, ProactiveXBench evaluates responsibility-conditioned intervention trajectories in physical scenarios. Agents infer latent intervention needs from evolving observations without explicit trigger conditions. *Timing*, *Progress*, and *Outcome* jointly assess initiation alignment, execution duration, and completion of ordered, entity-associated events, linking intervention discovery with the execution and completion of assistance.

D Prompts for Query-to-Identity Reformulation

This section presents the prompts used in the query-to-identity reformulation pipeline. The pipeline first determines whether a query-based task can be converted into a proactive task, then generates an identity-driven prompt that removes explicit triggering conditions while preserving the required intervention responsibility.

D.1 Task Conversion Prompt

The following prompt is used to determine whether an original query-based task can be transformed into a proactive task and to generate both explicit-event and identity-driven formulations.

Task Conversion Prompt

You are an expert annotator for proactive interaction training data. The input contains task descriptions, actions, questions, or guidance annotations from existing datasets. Determine whether the instance can be reliably converted into a proactive response task with a meaningful intervention time, and generate two formulations for the same intervention point.
Requirements:
1. **explicit_event_task**: Clearly describe the observable situation and the event that should trigger a response. The agent should wait before the event occurs and respond when the event happens. Do not treat the expected assistant response itself as a visual event. If only a response instruction is provided, rewrite it as the point where the task progress requires such guidance or correction.
2. **role_play_task**: Define a natural and specific assistant identity and context that allows the agent to decide when intervention is appropriate. Do not reveal the exact triggering event, original action description, or expected response content.
3. Both formulations should only describe when and why the agent should intervene, and should not generate the actual response.
4. Do not introduce facts unsupported by the original annotation. Mark convertible=false for greetings, generic praise, insufficient information, or cases where the intervention time cannot be determined.
5. Use concise and natural language. Return one JSON object for each input with the original id.
Output only the JSON array without Markdown, explanations, or reasoning.

D.2 Identity-Driven Role Prompt Generation

After identifying convertible samples, the following prompt is used to transform explicit query-based descriptions into identity-driven role prompts. The generated prompt only specifies the agent identity, general observation responsibility, and proactive intervention principle.

Role Prompt Generation Prompt

You are an annotator for proactive interaction training data. Given the original action descriptions, guidance instructions, and task categories, infer a specific professional role that can provide appropriate assistance and rewrite the task into a role-based prompt.
The role should be specific, such as a dance instructor, gardening assistant, fire safety trainer, cooking instructor, equipment maintenance specialist, or craft tutor. Avoid generic roles such as "task assistant" or "operation helper". The role name should represent expertise but should not contain the specific action being performed.
The generated prompt should only include: (1) the professional identity, (2) general observation responsibilities, and (3) the principle of providing proactive feedback when necessary.
The prompt must not reveal: - the specific task being performed, - concrete actions, - target objects or states, - triggering events, - execution steps, - expected responses, - intervention timestamps.
Do not write statements such as "observe the user performing [specific task]". Do not copy category labels. Do not claim that the input contains multiple views.