"Great! The Next Step Is...": In Pursuit of Proactive Assistance with Multimodal Foundation Models

Sean Andrist
Microsoft Research
Redmond, WA, USA
sandrist@microsoft.com

Dan Bohus
Microsoft Research
Redmond, WA, USA
dbohus@microsoft.com

Maia Stiber
Microsoft Research
Redmond, WA, USA
maiastiber@microsoft.com

Yuwei Bao
Microsoft
Redmond, WA, USA
baoemily@microsoft.com

Tim Schoonbeek
Eindhoven University of Technology
Eindhoven, Netherlands
t.j.schoonbeek@tue.nl

Eric Horvitz
Microsoft
Redmond, WA, USA
horvitz@microsoft.com

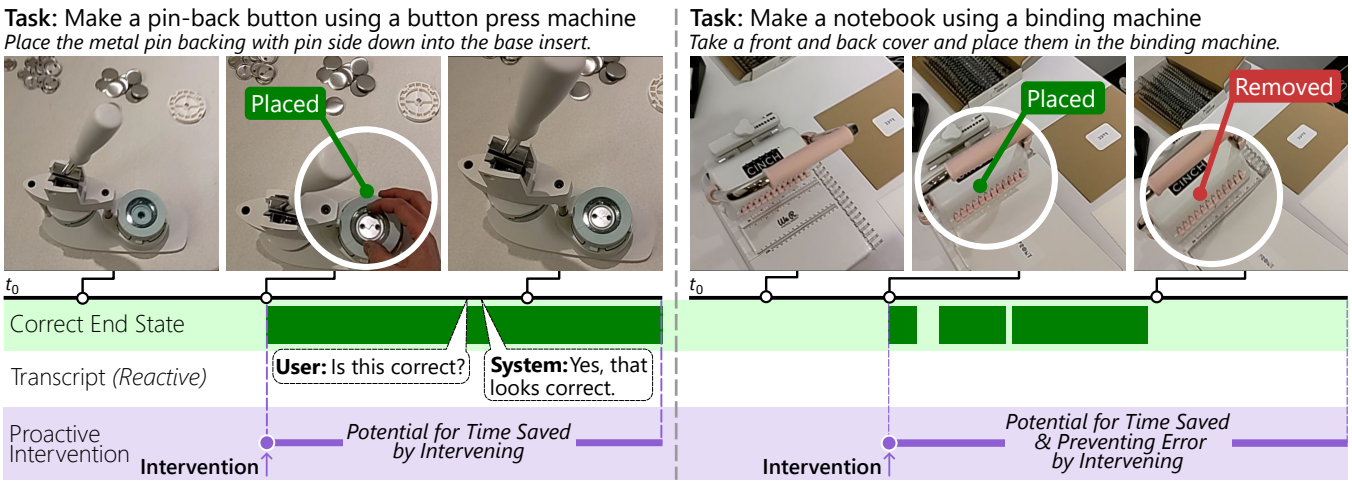


Figure 1: We explore how multimodal foundation models can be employed to potentially provide proactive assistance by enabling in-stream step completion detection using a variety of prompting techniques and evaluation metrics. This figure illustrates the potential for proactive step completion interventions to both save time and prevent errors.

Abstract

Large language and multimodal models have been largely employed to *reactively* generate content and assistance. In contrast, we explore how models can be employed to provide *proactive* assistance that fluidly supports ongoing human activities. We identify and tackle challenges in such *streaming scenarios*—where models must act continuously and responsively in real time, rather than waiting for explicit user requests. We explore solutions and evaluation metrics in the context of a mixed-reality AI agent designed to assist people in carrying out procedural physical tasks. Specifically, we focus on the challenge of identifying when each step of a procedure has been completed so that an agent can appropriately move on to the next step. Using a publicly available multimodal dataset of task-centric human-agent interactions, we study an array of prompting strategies across several frontier LLMs and VLMs, considering both the content and timing of model queries. Our findings demonstrate the criticality of developing application-driven metrics that enable meaningful comparisons and highlight key gaps in current model capabilities for proactive assistance. This work underscores the importance of temporally-aware evaluation in designing fluid real-time human-AI collaboration.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**.

Keywords

Proactive AI Agents, Procedural Task Assistance, Multimodal Foundation Models, Metrics

ACM Reference Format:
Sean Andrist, Dan Bohus, Maia Stiber, Yuwei Bao, Tim Schoonbeek, and Eric Horvitz. 2026. *"Great! The Next Step Is..."*: In Pursuit of Proactive Assistance with Multimodal Foundation Models. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION (ICMI '26), October 05–09, 2026, Napoli, Italy*. ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3776574.3831228

ICMI '26, Napoli, Italy

ACM ISBN 979-8-4007-2318-6/2026/10
https://doi.org/10.1145/3776574.3831228

1 Introduction

Frontier large language models (LLMs) and vision-language models (VLMs) are rapidly improving in their ability to assist humans across a wide range of tasks including answering complex questions, helping with coding, generating creative content, extracting insights from unstructured data, and multi-step planning. These models are particularly attractive due to the promise of zero-shot generalization which allows them to perform new tasks without task-specific training or fine-tuning [43]. As their capabilities continue to evolve, new possibilities are emerging for human-AI interaction in domains ranging from education and productivity to healthcare and robotics [9, 21, 40, 41]. In particular, procedural task assistance—guiding users step-by-step through physical activities—has surfaced as a compelling application area where multimodal understanding of objects, actions, and states is essential [31, 33, 39].

The majority of these models are inherently *reactive* in nature. They operate under a prompt-response or question-answer paradigm, where the model waits for a user's explicit (potentially multimodal) input before generating a response. While this chatbot paradigm is dominant today, foundation models hold significant potential to enable mixed initiative and *proactive* AI systems—agents that can take initiative, anticipate user needs, and act autonomously in the flow of activity, rather than waiting passively for instructions. Recent advances in streaming and real-time models further expand this space, enabling continuous processing of audio and video inputs [10, 11]. We use the term *streaming scenario* to refer to settings in which models must operate continuously over time, processing evolving multimodal context and making decisions in real time rather than responding to isolated queries. Such proactive systems have the potential to be more helpful, more efficient, and more natural in human-computer interaction, particularly in dynamic, real-world contexts such as physical task collaboration, assistive care, and emergency situations [20, 27, 29, 30, 32].

To effectively deploy prompt-based models in streaming scenarios, developers must at a minimum decide (1) which model(s) to use, and (2) how to prompt them, *i.e.*, how to populate prompts with relevant instructions and context. Designing for proactivity introduces additional, less discussed dimensions to consider. First is the question of *when* to prompt the model: on a fixed cadence over time or dynamically based on specific triggers or expectations associated with a particular task? Second is the question of how to evaluate the success of a proactive system—not just based on content quality (what it says), but on timing (when it says it). Saying the right thing is not enough; it must also be said at the right moment. For instance, proactively giving a user the next instruction too early can lead to confusion and errors; giving it too late can cause frustration and wasted time. Evaluating a proactive system's timing requires temporally aware and task-specific metrics that reflect user experience and task outcomes, not just query-level accuracy.

We investigate these questions through a concrete mixed-reality task assistance scenario by leveraging SigmaCollab [5], a publicly available dataset of human-AI collaboration during procedural task execution in the physical world. This dataset was collected using an open-source interactive system called Sigma [6, 7] in its purely reactive state; it waited for the user to explicitly signal task progress (*e.g.*, "I'm done, what's next?") rather than autonomously advancing when a step is completed [5]. To explore how a system could be made more proactive by delivering the next step instruction without requiring an explicit prompt from the user, we conducted a series of offline experiments using the SigmaCollab dataset. We tested a range of frontier LLMs and VLMs, evaluated diverse prompting configurations spanning multiple modalities of context and temporal sampling strategies, and assessed a battery of metrics from the generic to the application-specific. Figure 1 illustrates the potential benefits of proactive interventions in two examples from the dataset, including saving time and preventing errors.

We share important lessons learned and design implications for enabling assistive agents that leverage frontier LLMs and VLMs to proactively deliver task instructions based on detected completion. We demonstrate the need for multiple metrics to capture performance, particularly application-driven metrics like time savings and counting premature interventions, which are most impactful on the eventual user experience. Furthermore, we show that there is a large variance in the metrics across models and prompting strategies, with no clear winners and much more progress needing to be made before usability studies become realistic and informative. We highlight the nuanced design and inference considerations that make it important to give careful attention to the subtleties of the challenge—even for a seemingly "simple" problem like proactively moving to the next step. To support full reproducibility and future research, we publicly release all annotations created for this work, as well as all instantiated prompts and results across all models[1]. Ultimately, we argue that in-stream step completion detection is a necessary component of proactive procedural assistance, requiring careful temporal coordination and well-crafted, useful metrics.

2 Background and Related Work

2.1 Conceptual Foundations of Proactivity

We adopt the definition of *proactivity* proposed by Grosinger (2022), who describes it as "the ability to autonomously initiate anticipatory action based on reasoning, meant to impact people and/or their environment" [15]. Proactivity involves more than merely initiating actions in response to predicted human needs; it also encompasses the ability to deliberately *not* act when inaction is deemed appropriate [26].

Another source of theoretical grounding is provided by Berube *et al.* [4], who developed a conceptual model of proactivity for voice assistants with three components: *context*, *initiation*, and *actions*. Importantly, Berube *et al.* also highlights a gap in literature by noting only five studies (of all that were considered) employed operational prototypes of proactive voice assistants, while most relied on Wizard-of-Oz methods [4]. This observation underscores the need for more application-driven research.

Prior efforts in human-computer interaction, including work in the realm of situated interaction, have explored designs, principles, and machinery for guiding mixes of human and machine initiative, centering on the *if, when, and how* of human and machine contributions during human-AI collaborations [8, 18]. These foundational concepts around mixed initiative inform our approach to proactive step completion detection, where the timing and appropriateness of system-initiated actions are central considerations.

[1] https://aka.ms/sigma-correct-state

2.2 Implementations of Proactive Agents

Research has begun shifting the focus from reactive AI assistants to proactive systems that preemptively support users by understanding changing contexts and anticipating needs, rather than merely reacting to explicit verbal commands or other direct inputs (*e.g.*, button presses or touch). However, much of the existing work still operationalizes proactivity in a chatbot paradigm with rigid turns, rather than fluidly in stream. ProactiveBench [24] is a benchmark proposed to elicit proactiveness in LLM-based agents. [12] and [28] explore proactive agents that offer timely coding suggestions based on user activity, while evaluating the tradeoffs between efficiency and cognitive disruption. Moving beyond language-only interfaces, embodied agents such as AssistantX [36], FlowAct [13] and others [27] demonstrate how real-time interactive agents can proactively respond within physical environments. More closely related to our work, multimodal systems like Satori [23] and YETI [2] leverage augmented reality (AR) and egocentric vision to model user intention, predict actions, and detect opportunities for proactive guidance in procedural tasks. Collectively, these studies emphasize the timing and effectiveness of guidance during human-AI interaction.

Taken together, these perspectives illustrate how proactivity manifests across modalities, interfaces, and interaction paradigms. In this work, we conceptualize proactivity as an agent's ability to initiate—or deliberately withhold—timely interventions grounded in an application-driven, real-time multimodal context, with a particular emphasis on procedural tasks in physical environments. Our focus on detecting step completion in-stream is related to, but distinct from, prior work on action recognition and procedure step recognition in computer vision [31, 33]. While those works focus on recognizing actions and handling execution errors from egocentric video using task-specific trained classifiers, our work centers on the real-time decision of *when to intervene* in an ongoing human-AI collaboration, using zero-shot foundation models. We contribute to this growing body of research by evaluating the in-stream timing of decisions to deliver the next task instruction, and introducing a set of metrics to assess impact on potential user experience. This sets the stage for a deeper examination of how proactive behaviors can be designed, operationalized, and evaluated in real-world contexts.

2.3 Evaluating Proactive Agents

Traditional task-driven evaluations primarily assess how AI assistance impacts task success rates and the number of steps or dialogue turns needed to achieve a goal [2, 19, 23, 38, 42]. Another important dimension evaluates the quality of language outputs, using conventional metrics such as BLEU, ROUGE, GLEU, and METEOR, as well as more recent LLM-assisted evaluations that assess groundedness, relevance, coherence, fluency, grammatical correctness, and semantic similarity [14, 17, 34]. Recent work introduces comprehensive frameworks to systematically evaluate the capabilities of LLMs and VLMs across these dimensions [1]. Complementary studies have incorporated human evaluations or Wizard-of-Oz setups [2, 3, 16, 37], measuring factors such as perceived helpfulness, satisfaction, annoyance, timing, disruption, and trust.

As AI assistants become increasingly proactive, evaluating the quality and timing of their interventions is crucial. Traditional approaches [2, 25, 27, 39] rely on confusion-matrix-based metrics (*e.g.*, F1 score, precision, recall) and on segment-based metrics against ground-truth time spans to assess the correctness and timeliness of interventions. Another study [31] evaluates model timeliness by measuring the average absolute delay between the ground-truth completion of a procedural step and subsequent recognition of that step. Real-time interactive systems must consider system-level metrics such as response latency, frequency of interventions, and the accuracy of triggers for proactive actions [2, 12, 24, 35].

In this work, we introduce an evaluation framework for real-time, in-stream decision-making utilizing an application-driven dataset. Conventional model-centric metrics often obscure interaction-level impact: a single misprediction among hundreds of frames may be statistically negligible from a model-centric perspective, yet trigger catastrophic results during an interaction. Likewise, late predictions provide reduced positive benefit, and too-early predictions can create confusion or even induce human error—an observation consistent with prior work on the effects of input errors on user experience [22]. We, therefore, focus on two complementary metrics: (i) time saved as a percentage of the total optimal savings, and (ii) a strict penalty for false positives reflecting their potentially undefined consequences in interactive systems. By grounding evaluation in these metrics, we align performance assessment with interaction-driven objectives rather than abstract model accuracy.

3 Problem Definition

In this paper, we focus on the problem of *in-stream step completion detection* in a physical task assistance scenario. *In-stream step completion detection* would enable an interactive AI system to determine in stream (*i.e.*, without access to the future and as early as possible) when the user has correctly completed each step in a procedure. The system could then take the initiative at that moment and proactively deliver the next instruction. As a first step towards enabling this capability, we can first identify traces of step executions when a human collaborates with a reactive AI system, distinguishing intervals of in-progress work vs end state achievement. Inspection and analysis of the SigmaCollab dataset [5] reveals that formalizing this problem by adopting the perspective of the worker's actions, *i.e.*, temporally annotating the answer to the question: "has the worker finished performing the instructed action correctly?" is challenging. Workers may perform multiple actions simultaneously; they may perform actions atypically (*e.g.*, lifting the water tank off the coffee machine before filling it with water); they may linger and produce micro-movements (*e.g.*, placing a cup on the drip tray but then nudging it slightly); and so forth.

A better formalization is to instead adopt a world-state perspective, focusing on the task-relevant objects and their relationships, and answering the following question: "is the world in the correct state expected if this step (and all previous ones) were performed correctly?" Reasoning over objects and their states has been well studied in computer vision and human-robot interaction [31, 33]; our contribution is to apply this perspective as a proxy for *in-stream decision-making* about when a system could proactively intervene, rather than for post-hoc action recognition. This problem is illustrated in Figure 1, where intervals of the world being in the "correct end state" are depicted, with timely intervention points showing the potential to save time in task execution, eliminate the need for

further questions and confirmations, and even prevent errors when users might mistakenly break a correct state after achieving it.

4 Method

To explore how multimodal foundation models might be leveraged in a system for proactive instruction-giving, we conducted offline experiments with several models and prompting strategies using the SigmaCollab dataset [5] within the context of in-stream step completion detection. This dataset is comprised of multimodal recordings (including the egocentric images in Figure 1) of participants executing procedural tasks while interacting with a mixed-reality agent instructor [6] that required explicit, verbal commands to move onto the next step. This behavior contrasts with the fluid step-by-step transitions observed in most human-human collaborations (*e.g.*, in the HoloAssist dataset [39], the *human* instructor anticipates the right time to deliver the next instruction without the worker usually needing to say anything). SigmaCollab also contains *expert demonstrations* for each task, providing a resource for prompting models with prior knowledge on similar step executions.

4.1 Ground Truth Annotations

Two annotators (two of the authors) marked correct end state temporal intervals in all sub-step executions, each covering half of the existing SigmaCollab dataset. During the annotation process, each annotator marked instances of ambiguity or annotation uncertainty. The annotators then came together to resolve these ambiguities in an iterative and collaborative process, until an adequate consensus was judged to have been achieved.

The annotators adopted a world-state perspective, rather than action-centric, focusing on object and object-part relationships (not worrying about actions and hand movements). The goal was to identify time intervals in which the state of the objects corresponded to the *reference* world state at the end of a fully correct execution of the step. During the time periods when the step was not completed yet, or a mistake was made, or the worker moved past the current step to a future step already, the world is not considered to be in the correct end state. As a result, within the span of a single step, there can be one correct end state annotation interval, multiple such intervals (*e.g.*, a worker could undo and redo their correct action), or none at all (see Figure 1 and 2). Finally, to make the ground truth annotations as objectively accurate as possible, the annotators were allowed to leverage future information to identify and mark correct end state intervals in the past.

Nevertheless, many subtleties still remain in the data. For example, in certain steps, such as muddling the mint at the bottom of a shaker, or tightening a screw by hand, the precise moment when the world enters the correct end state (*i.e.*, the mint is muddled enough, the screw is tightened), is not immediately ascertainable from the available egocentric views. Other steps involve pouring certain amounts of liquids or ingredients, where again there are degrees of subjectivity into what would be "not enough" or "too much." We established a set of conventions for these type of cases, such as relying on observing when the worker has stopped performing the action (we trust the worker) in the former case, or tolerating the pouring of somewhat variable amounts (as long as the correct liquid was used) if the pour quantity could not be determined visually.

At the same time, the annotation schema did require that the worker conformed to the language of the instruction given, and object substitutions were generally not permitted. For instance, if a step in building a drink called for adding mint leaves into a "Boston shaker" and the worker used instead a "cobbler shaker," we consider the step to have never entered the correct end state, even if the drink contents were correct. If the worker continued building the drink in the cobbler shaker, that also meant all subsequent steps never entered the correct end state.

Following the ground truth annotation process, each sub-step was labeled for its *overall* correctness. If the sub-step ends with a correct end state interval, it is marked as *correct*, otherwise it is marked as *incorrect*. Sub-steps were also labeled as *interrupted* if the experimenter intervened, the worker abandoned the task, or the system crashed during the course of the sub-step. If there was no adequate visual evidence during the sub-step about when it was correctly completed (*e.g.*, if the participant was wearing the headset at an odd angle and the camera was pointed in an uninformative direction), then the sub-step was marked as *unknown*.

4.2 Temporal Prompting Strategies

There are many possible approaches for prompting large multimodal models to solve an in-stream classification problem such as the *correct end state detection* problem defined above. We explored three such strategies in this work. The actual models and prompt contents we tried are explained in Sections 4.4 and 4.5.

4.2.1 Fixed-Cadence Prompting. One straightforward approach to prompting foundation models in stream is to query them at a fixed cadence, probing the model every t seconds. The prompts can be filled in with multimodal context over time, such as the time elapsed, current POV image, any dialogue that has taken place so far, etc. In this work, we explore prompting at 1Hz.

When used for proactive task guidance, a policy can be fit on top of these model results to determine a *decision point*: the timepoint at which the system decides to proactively move to the next instruction. A simple "immediate" policy would be to set this decision point at the first time the model returns "yes" (the world is in the correct end state). More conservative policies might wait for multiple consecutive "yes" answers, and even more sophisticated policies can be imagined involving estimated confidence signals.

4.2.2 One-Time Prompting. Another even simpler (and cheaper) prompting approach is to query a foundation model only a single time, at the very beginning of each sub-step, to estimate how long that sub-step should take to complete. Whatever answer the model returns corresponds to the decision point. For example, if the model predicts 15 seconds, then the decision point is 15 seconds after the start of the sub-step. This strategy provides less of an opportunity to fill in dynamic context into these prompts (*e.g.*, it does not have access to any dialogue that might happen between the worker and the system during sub-step execution).

4.2.3 Dynamic Prompting. Finally, we explored a temporally dynamic extension to the one-time prompting approach. In the dynamic approach, the model is queried at the beginning of the sub-step as before, to predict how long the sub-step should take to complete. Once that time has elapsed, the model is queried again

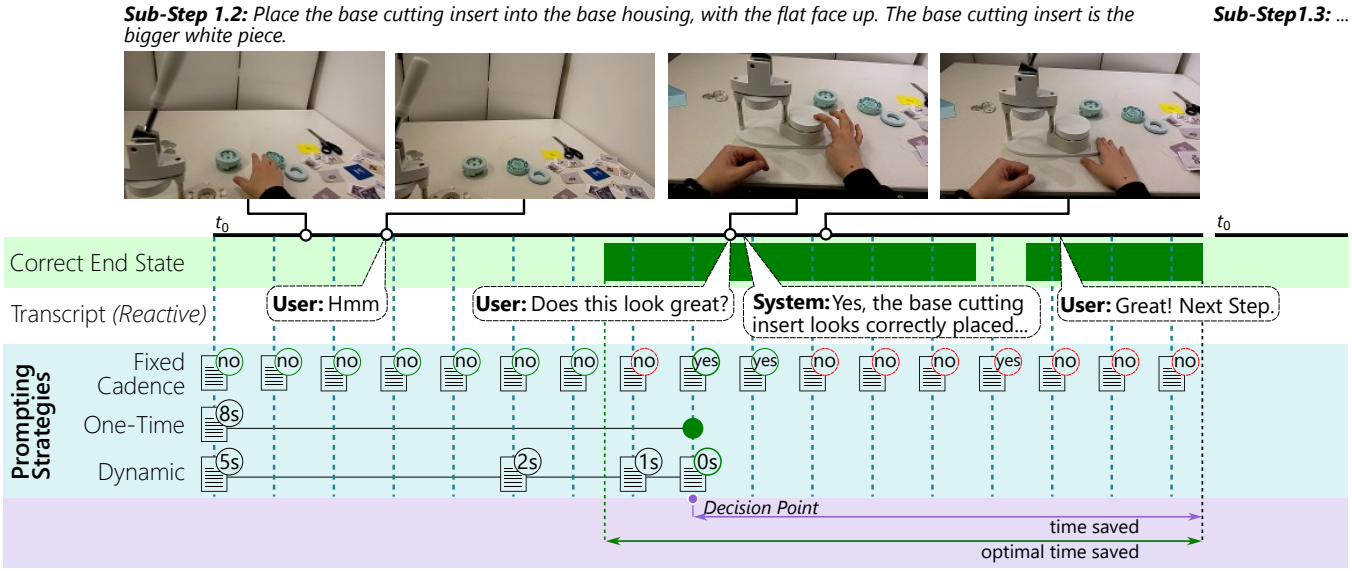


Figure 2: Illustration of temporal prompting strategies and metrics. For the *fixed cadence* prompting strategy, a query is issued every one second, and the model is asked to answer "yes" or "no". For the *one-time* strategy, the query is issued only at the beginning of the sub-step and returns a number (in this case 8 seconds). For *dynamic* queries, the model returns estimated times after which it checks again, until it determines that the sub-step is complete by returning 0 seconds. In this illustration, the decision points are at the same timepoint for all three prompting strategies (although in reality they could all be different).

(with additional context extracted from the elapsed interaction) to predict whether the sub-step is indeed now complete, or if more time is needed. The model thus has the opportunity to take into account evolving multimodal context, such as the camera images and dialogue history, but greatly reduces the computation cost by generating new queries in a dynamic, self-guided manner.

4.3 Evaluation Metrics

Given the prompting strategies above and the problem definition, several metrics are available for evaluating performance across models and prompt contents.

4.3.1 Query-level Metrics. [Accuracy and *F1 score]*

The initial approach that machine learning practitioners might instinctively take is to treat this problem as a basic classification paradigm and compute query-level metrics. Each query of a particular prompt instantiation is evaluated independently as either a true positive, false positive, true negative, or false negative, according to its prediction value and whether it is located within a ground truth correct interval. An example of how to compute query-level accuracy metrics is shown in Figure 2: the model produces 7 true negatives, 2 true positives, 7 false negatives, and 1 false positive.

Although metrics like accuracy and F1 are then straightforward to compute from these query-level results, they might not reflect the actual perceived performance of the given model if deployed at runtime to trigger a proactive move to the next step [22]. What really matters is the *decision point*, *i.e.*, the earliest time the AI assistant believes a proactive intervention is needed. For the one-time prompts, this decision point is at the time predicted by the model at the beginning of the step. For fixed cadence and dynamic prompts, the decision point is the first time that the model predicts "yes" the sub-step is complete (Figure 2).

4.3.2 Step-level Metrics. [Time Saved and *False Positives]*

To better capture actual impact on the worker's experience and application performance, we focus on the model's decision point and its impact within each sub-step. For example, if the decision point falls within a correct end state ground truth interval, that prediction can be marked as a *step-level* true positive, regardless of any predictions after (see Figure 2). Early true positive decision points are clearly better than later ones. If the decision point falls outside the correct interval, it is considered a step-level false positive. If the model never says "yes" (no decision point), it is a step-level false negative if there are any correct ground truth intervals, and a true negative otherwise.

Percentage of Optimal Time Saved: For true positives, defined as *time saved* divided by the *optimal time saved*. *Time saved* is the time between the decision point and the end of the step. *Optimal time saved* is the time between the start of the first correct interval and the end of the step. The reason for this metric is that simply counting step-level true positives does not tell the whole story of how successful the model's proactive behavior would be in practice. A true positive at the beginning of the correct end state interval is more desirable than a true positive at the end of the interval. An early true positive might even prevent a later mistake if the worker is not sure that they are in the correct end state and were to accidentally break it later on (see right half of Figure 1). In Figure 2, the *time saved* is 8.5 seconds, while the *optimal time saved* is 10 seconds before the end of the sub-step. Thus, the *percentage of optimal time saved* in this example would be 8.5 / 10 = 85%.

False Positive Percentage: Defined as the ratio of false positives to total samples (sub-steps). False positives can be extremely damaging to the overall interaction, leading to confusion, mistakes, and potential task failure. They are the most costly error made by a model, and therefore most important to pay attention to, as they correspond to the system moving onto the next step when the worker is not done with the current step.

4.4 Models

We tested five frontier multimodal foundation models: Llama-3.2-Vision, Qwen3.5, GPT-5.1, GPT-5.4, and Gemini-3.1-Pro. These models span a range of architectures, providers, and capability levels, and all accept both text and image inputs. We chose models with reasonably fast inference times and low latencies to support eventual integration into a live system prompting at up to 1 Hz, and therefore excluded slower reasoning models. Actual compute latencies were not considered in this offline experiment.

Additionally, we included a condition of incrementally prompting Gemini-3.1-Pro with video clips of the sub-step execution up to that point. This approach allows us to compare a model that can natively reason over temporal visual sequences rather than individual image frames.

4.5 Prompt Contextual Contents

We crafted several kinds of prompts with varying categories and degrees of context, across all three temporal strategies. All prompt templates are fully detailed in Appendix A.

All prompts included a description of the physical task the human worker is performing, the specific sub-step they are working on at that moment, and the status of the system's instruction-delivering utterance (just beginning, in progress, or fully spoken). The prompt was filled in with varying amounts of context, and the model was then instructed on what it was supposed to infer. For the fixed-cadence prompts, the model was essentially instructed to answer "yes" or "no" to the question: "based on the provided context, predict whether or not the human worker is likely to have completed the current step by now". For the one-time and dynamic prompts, the model was instructed to provide the number of seconds the human worker likely will need to complete the sub-step (could be 0).

The different prompt templates we tested in the fixed-cadence experiments, involving differing categories of context, are listed below. Each corresponding prompt was instantiated and filled in with dynamic data over time. All fixed-cadence prompts include the subdialog context (a transcript of any dialog that has taken place between the system and the worker so far in the current sub-step) as a baseline context layer.

- **Subdialog**: The baseline prompt, which includes the subdialog transcript and the amount of time that has elapsed in the sub-step so far, in seconds.
- **Reading**: In addition to the subdialog context, these prompts include a timestamped list of intervals when the participant was likely reading the task instructions from the virtual task panel hologram, rather than looking at the physical task space. We generated these annotations automatically by intersecting the worker's gaze ray with the location of the floating user interface rectangle holograms in 3D space, eliminating intervals shorter than 0.5 seconds, and merging intervals with gaps of less than 0.5 seconds.
- **Image**: In addition to the subdialog context, these prompts include the egocentric camera view at that moment. We used the right-front-facing grayscale image because it had a better vertical field-of-view to capture the task space than the RGB images.
- **All (Subdialog + Reading + Image)**: These prompts combine the subdialog context with both the reading annotations and the egocentric camera view.
- **Image vs Expert Reference Image**: These prompts include the subdialog context, the egocentric camera view of the worker at that moment, and for comparison: the egocentric camera view of an expert who just completed the sub-step, and the time it took that expert to do so.

For the one-time prompting strategy, we implemented a simple prompt that only described the task and sub-step instruction under consideration. For the dynamic prompt, we progressively filled in the prompt contents with subdialog, reading state, and egocentric images, as in the *All* fixed-cadence prompt.

We also compared all models and prompts to a simple parameterized baseline derived from the expert demonstrations. The baseline $B(n)$ was defined by predicting that each sub-step will take the same amount of time it took the expert to complete the same sub-step, plus n additional seconds for various assignments of n. Intuitively, this baseline captures a "best-case" timing heuristic: if an expert's completion time is known, we simply add a buffer of n seconds to account for a novice worker's likely slower pace. The parameter n allows us to sweep from aggressive ($n = 0$, assuming the worker is as fast as the expert) to conservative ($n >> 0$) predictions. This simple baseline is surprisingly competitive with the frontier models, as the results below demonstrate.

4.6 Results

We instantiated and tested prompts on the entire dataset, across all of the prompt categories and models listed above. However, we only considered sub-steps that *start* correctly (*i.e.*, the previous sub-step ended correctly, or it is the first step of the recipe), regardless of whether the sub-step *ends* correctly or incorrectly. We also excluded any steps that were marked as *unknown* or *interrupted*, leaving 1,179 sub-steps for the evaluation. Sub-steps ranged in length from 3 seconds to 7 minutes 52 seconds. For fixed-cadence prompting, this resulted in 32,304 query instances per model/prompt combination across the dataset.

A subset of the most notable results can be seen in Table 1. Llama, Qwen, and One-Time prompts are excluded for space (and low performance), but the full table of results is listed in Appendix B. Gemini-3.1-Pro with fixed-cadence video prompts performed the best in terms of accuracies and F1 scores, as well as percentage of optimal time saved. However, it also exhibits a very high rate of damaging step-level false positives, showcasing the need to consider multiple application-driven metrics, and the inadequacy of query-level metrics alone. The dynamic prompting approach (again Gemini-3.1-Pro) seems to have the best balance of low false positives with meaningful time saved.

Table 1: Performance across prompt versions and models. Best performance values for each metric are marked in bold.

Prompt Version	Model	Accuracy (Query)	F1 (Query)	Accuracy (Step)	F1 (Step)	False Positive Percentage (Step)	Percent Optimal Time Saved
Baseline(0)	N/A	N/A	N/A	36.5	0.518	39.6%	35.8%
Baseline(+10s)	N/A	N/A	N/A	16.7	0.232	18.7%	23.6%
Image	GPT-5.1	77.0	0.363	27.8	0.396	37.4%	32.6%
	GPT-5.4	78.0	0.312	32.8	0.456	20.4%	32.3%
	Gemini-3.1-Pro	81.3	0.544	45.0	0.604	35.8%	40.1%
All Context	GPT-5.1	75.6	0.355	25.2	0.368	44.3%	29.7%
	GPT-5.4	77.3	0.267	29.0	0.407	22.3%	26.0%
	Gemini-3.1-Pro	81.6	0.554	48.2	0.636	35.7%	44.9%
Video	Gemini-3.1-Pro	**82.6**	**0.655**	**50.0**	**0.655**	47.9%	**45.3%**
Dynamic	GPT-5.1	N/A	N/A	26.7	0.398	48.8%	27.7%
	GPT-5.4	N/A	N/A	28.0	0.398	32.3%	26.9%
	Gemini-3.1-Pro	N/A	N/A	39.3	0.530	**11.0%**	31.9%

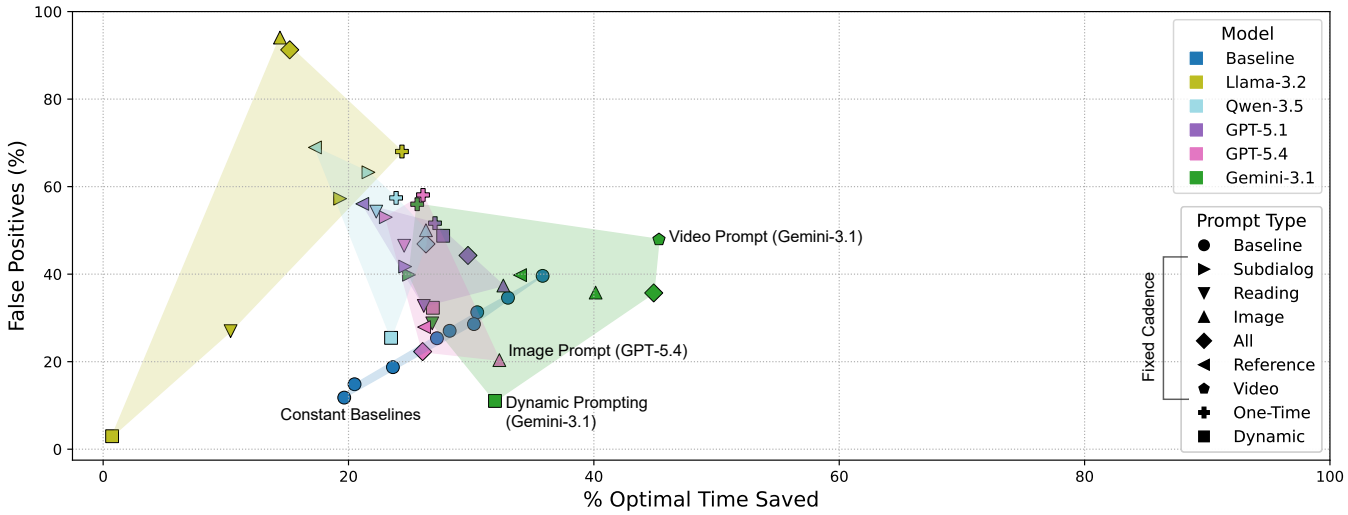


Figure 3: Percentage of optimal time saved plotted against step-level false positive percentage. Colored contours around model.

We plot performance on the two application-driven metrics in Figure 3: percentage of optimal time saved and false positive percentage. Optimal performance is towards the bottom-right of the plot. This figure illuminates the high difficulty of the problem, with a large gap in time savings still achievable while keeping false positives low. In fact, very few approaches beat the baselines of simply predicting the expert demonstration step length + n. In terms of fixed cadence prompting, GPT-5.4 with image prompts is competitive, with reasonable time savings and 20.4% false positives. Prompts to Gemini-3.1-Pro seem to do the best overall, with more results lying to the bottom-right. The dynamic prompting approach to Gemini seems particularly attractive.

Figure 4 illustrates the impact of different decision strategies on prompt performance. Using the video and image prompts as examples, performance is plotted when proactive decision points are considered immediately on the first "yes" prediction (as above), vs requiring two or three consecutive positive predictions in a row. These more conservative strategies can result in fewer false positives, potentially at the expense of aggressively saving time.

5 Discussion and Limitations

Our results reveal a disconnect between conventional query-based metrics and task-driven, application-level metrics. While query-level accuracy is commonly reported in the literature, it does not reliably predict which model or prompt will produce the best user experience in real-time, proactive interactions. Metrics that capture temporal and interactive dimensions—such as time saved and the frequency of premature interventions—are therefore critical for evaluating the effectiveness of proactive agents. Evaluating models solely on standard benchmarks can be misleading, as performance varies widely across models and prompting strategies.

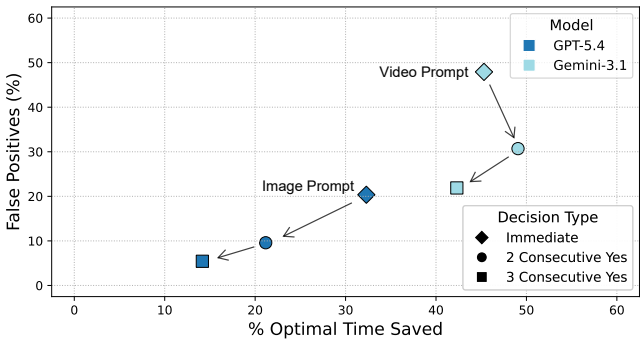


Figure 4: Impact of different decision strategies.

We treat the classification of "correct end state" over time as a proxy for proactive assistance. While useful for benchmarking, this approach does not fully determine when or how an agent should act. Real-time intervention requires reasoning about cost/benefit tradeoffs, handling uncertainty, accounting for inference latency, and coordinating with the user's ongoing behavior (e.g., attention and speech in progress). Automatically advancing to the next step is only one potential policy; an agent could instead ask "are you ready?", defer when speech and gaze suggests distraction, or require multiple cues. Policies governing the agent's aggressiveness—deciding when to move the user forward, when to check readiness, and when to wait for more evidence—remain open challenges. One promising direction is to design interactive systems where the model is only tapped to proactively engage when the false positive rate is low. For example, an agent could operate in a conservative mode by default, only triggering a proactive intervention when multiple signals converge (*e.g.*, visual evidence of step completion, absence of ongoing user speech, and elapsed time consistent with expected step duration). Such selective engagement could mitigate the cost of false positives while still capturing the time-saving benefit, and represents a practical path toward deployable proactive assistance.

On the role of offline evaluation. An important limitation is that these experiments were offline, rather than involving a live user study to validate the effectiveness of proactive interventions. We argue that at the current stage of model capabilities, a rigorous offline evaluation is not only appropriate but essential as a first step. The results presented here demonstrate that zero-shot frontier models still exhibit high variance and substantial false positive rates across prompting strategies. Deploying a system with these error rates in a live study would risk confounding the evaluation of proactive *timing* with user frustration arising from frequent erroneous interventions. An offline evaluation allows us to systematically characterize the design space—identifying which models, prompts, and temporal strategies show the most promise—before committing to the cost and complexity of a live deployment. Once models and techniques achieve reliably low false positive rates, a user study becomes both more feasible and more informative, and we consider this an important direction for future work.

Broader scope of proactivity. In this paper, we focus on one specific form of proactive behavior: step completion detection. We acknowledge that full proactivity encompasses a broader range of behaviors, including error correction, anticipatory guidance, clarification requests, and confirmations [4], requiring broader reasoning over intent, attention, utility, and interruption cost. That said, step completion detection serves as a foundational building block—if a system cannot reliably detect when a step is done, it is unlikely to perform well on more complex anticipatory behaviors. Future work should expand the scope of proactive interventions studied, building on the evaluation framework and lessons learned here.

Several additional limitations highlight avenues for future work. For example, this paper does not propose new modeling architectures or algorithms; rather, it explores considerations for evaluating proactive assistance and understanding the nuances of real-world task performance. Future work should also carefully handle "incomplete" steps, differentiating genuine mistakes from cases where users intentionally move ahead.

Safe and Responsible Innovation Statement. Proactive systems with next-step detection have the potential to provide adaptive support tailored to each person's needs and pace, benefiting individuals with diverse abilities. However, a system that requires continuous processing of images and/or gaze behavior raises privacy concerns around capturing environments and behavior patterns. Additionally, without ethical guidelines in place, these systems could be repurposed for workplace surveillance and used to pressure workers to increase their pace, elevating risk and stress. Finally, the SigmaCollab dataset did not appear to include underrepresented diverse populations in ability and background, potentially biasing model evaluations. Responsible deployment should ensure informed user consent, transparent data handling, inclusive data collection, and awareness against surveillance-focused misuse.

6 Conclusion

We investigated how large multimodal models can support proactive assistance in real-time human-AI interactions. Using the SigmaCollab dataset [5], we focused on detecting in-stream step completion to determine timely moments for delivering the next instruction, evaluating a range of frontier LLMs and VLMs.

Our work contributes several key insights. First, we demonstrate that enabling proactivity with today's foundation models requires careful attention to when models are prompted, not only how they are prompted or which model is used. Second, we introduce a system-driven evaluation framework that distinguishes between query-level and step-level metrics, showing that conventional accuracy-based evaluations can be misleading when applied to interactive, time-sensitive systems. Evaluating prediction quality jointly with timeliness offers a more faithful measure of the user's experience and the system's effectiveness. Third, we frame with this work a set of research directions forward in pursuit of detecting correct task completion in-stream—an essential building block for proactive, embodied AI systems. Fourth, we release all annotations, instantiated prompts, and model results to support full reproducibility and future research. Taken together, this work lays a foundation for designing, benchmarking, and reasoning about proactive multimodal agents that act fluidly, helpfully, and responsibly alongside people in the physical world.

Acknowledgments

The authors would like to thank Ann Paradiso, Nick Saw, Andy Wilson, and the rest of the Future Experiences group at Microsoft Research for their feedback and support.

GenAI Usage Disclosure

During the preparation of this work, the authors used Generative AI tools in order to improve readability and language of the final draft. All research contributions, results, and interpretations are the sole work of the authors. The authors take full responsibility for the content of the publication.

References

[1] Vidhisha Balachandran, Jingya Chen, Neel Joshi, Besmira Nushi, Hamid Palangi, Eduardo Salinas, Vibhav Vineet, James Woffinden-Luey, and Safoora Yousefi. 2024. Eureka: Evaluating and Understanding Large Foundation Models. arXiv:2409.10566 [cs.LG] https://arxiv.org/abs/2409.10566

[2] Saptarashmi Bandyopadhyay, Vikas Bahirwani, Lavisha Aggarwal, Bhanu Guda, Lin Li, and Andrea Colaco. 2025. YETI (YET to Intervene) Proactive Interventions by Multimodal AI Agents in Augmented Reality Tasks. arXiv:2501.09355 [cs.AI] https://arxiv.org/abs/2501.09355

[3] Yuwei Bao, Keunwoo Yu, Yichi Zhang, Shane Storks, Itamar Bar-Yossef, Alex de la Iglesia, Megan Su, Xiao Zheng, and Joyce Chai. 2023. Can Foundation Models Watch, Talk and Guide You Step by Step to Make a Cake?. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12325–12341. doi:10.18653/v1/2023.findings-emnlp.824

[4] Caterina Bérubé, Marcia Nißen, Rasita Vinay, Alexa Geiger, Tobias Budig, Aashish Bhandari, Catherine Rachel Pe Benito, Nathan Ibarcena, Olivia Pistolese, Pan Li, et al. 2024. Proactive behavior in voice assistants: A systematic review and conceptual model. *Computers in Human Behavior Reports* 14 (2024), 100411.

[5] Dan Bohus, Sean Andrist, Ann Paradiso, Nick Saw, Tim Schoonbeek, and Maia Stiber. 2025. SigmaCollab: An Application-Driven Dataset for Physically Situated Collaboration. *arXiv preprint arXiv:2511.02560* (2025).

[6] Dan Bohus, Sean Andrist, Nick Saw, Ann Paradiso, Ishani Chakraborty, and Mahdi Rad. 2024. SIGMA: An Open-Source Interactive System for Mixed-Reality Task Assistance Research. arXiv:2405.13035 [cs.HC] https://arxiv.org/abs/2405.13035

[7] Dan Bohus, Sean Andrist, Nick Saw, Ann Paradiso, Ishani Chakraborty, and Mahdi Rad. 2024. SIGMA: An Open-Source Interactive System for Mixed-Reality Task Assistance Research – Extended Abstract. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 889–890. doi:10.1109/VRW62533.2024.00241

[8] Dan Bohus and Eric Horvitz. 2011. Decisions about turns in multiparty conversation: From perception to action. In *Proceedings of the 13th international conference on multimodal interfaces*. 153–160.

[9] Alexia Cambon, Brent Hecht, Ben Edelman, Donald Ngwe, Sonia Jaffe, Amy Heger, Mihaela Vorvoreanu, Sida Peng, Jake Hofman, Alex Farach, et al. 2023. Early LLM-based tools for enterprise information workers likely provide meaningful boosts to productivity. *Microsoft Research. MSR-TR-2023* 43 (2023).

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

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

[12] Valerie Chen, Alan Zhu, Sebastian Zhao, Hussein Mozannar, David Sontag, and Ameet Talwalkar. 2025. Need Help? Designing Proactive AI Assistants for Programming. arXiv:2410.04596 [cs.HC] https://arxiv.org/abs/2410.04596

[13] Timothée Dhaussy, Bassam Jabaian, and Fabrice Lefèvre. 2025. FlowAct: A Proactive Multimodal Human-robot Interaction System with Continuous Flow of Perception and Modular Action Sub-systems. arXiv:2408.15864 [cs.RO] https://arxiv.org/abs/2408.15864

[14] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. arXiv:2306.13394 [cs.CV] https://arxiv.org/abs/2306.13394

[15] Jasmin Grosinger. 2022. On Proactive Human-AI Systems.. In *AIC*. 140–146.

[16] Ken Gu, Madeleine Grunde-McLaughlin, Andrew M. McNutt, Jeffrey Heer, and Tim Althoff. 2024. How Do Data Analysts Respond to AI Assistance? A Wizard-of-Oz Study. arXiv:2309.10108 [cs.HC] https://arxiv.org/abs/2309.10108

[17] Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. LLM-Rubric: A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 13806–13834. doi:10.18653/v1/2024.acl-long.745

[18] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 159–166. doi:10.1145/302979.303030

[19] Zhiyuan Hu, Yue Feng, Yang Deng, Zekun Li, See-Kiong Ng, Anh Tuan Luu, and Bryan Hooi. 2023. Enhancing Large Language Model Induced Task-Oriented Dialogue Systems Through Look-Forward Motivated Goals. arXiv:2309.08949 [cs.CL] https://arxiv.org/abs/2309.08949

[20] Rabeya Jamshad, Arthi Haripriyan, Advika Sonti, Susan Simkins, and Laurel D. Riek. 2024. Taking Initiative in Human-Robot Action Teams: How Proactive Robot Behaviors Affect Teamwork. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction* (Boulder, CO, USA) *(HRI '24)*. Association for Computing Machinery, New York, NY, USA, 559–562. doi:10.1145/3610978.3640640

[21] Callie Y Kim, Christine P Lee, and Bilge Mutlu. 2024. Understanding large-language model (llm)-powered human-robot interaction. In *Proceedings of the 2024 ACM/IEEE international conference on human-robot interaction*. 371–380.

[22] Ben Lafreniere, Tanya R. Jonker, Stephanie Santosa, Mark Parent, Michael Glueck, Tovi Grossman, Hrvoje Benko, and Daniel Wigdor. 2021. False positives vs. false negatives: The effects of recovery time and cognitive costs on input error preference. In *The 34th annual ACM symposium on user interface software and technology*. 54–68.

[23] Chenyi Li, Guande Wu, Gromit Yeuk-Yin Chan, Dishita G Turakhia, Sonia Castelo Quispe, Dong Li, Leslie Welch, Claudio Silva, and Jing Qian. 2025. Satori: Towards Proactive AR Assistant with Belief-Desire-Intention User Modeling. arXiv:2410.16668 [cs.HC] https://arxiv.org/abs/2410.16668

[24] 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. 2024. Proactive Agent: Shifting LLM Agents from Reactive Responses to Active Assistance. arXiv:2410.12361 [cs.AI] https://arxiv.org/abs/2410.12361

[25] Esteve Valls Mascaro, Hyemin Ahn, and Dongheui Lee. 2024. Intention-Conditioned Long-Term Human Egocentric Action Forecasting. arXiv:2207.12080 [cs.CV] https://arxiv.org/abs/2207.12080

[26] Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. 2024. When to show a suggestion? Integrating human feedback in AI-assisted programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 10137–10144.

[27] Maithili Patel and Sonia Chernova. 2022. Proactive Robot Assistance via Spatio-Temporal Object Modeling. arXiv:2211.15501 [cs.RO] https://arxiv.org/abs/2211.15501

[28] Kevin Pu, Daniel Lazaro, Ian Arawjo, Haijun Xia, Ziang Xiao, Tovi Grossman, and Yan Chen. 2025. Assistance or Disruption? Exploring and Evaluating the Design and Trade-offs of Proactive AI Programming Support. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. ACM, 1–21. doi:10.1145/3706598.3713357

[29] Tyler Reinmund, Lars Kunze, and Marina Denise Jirotka. 2024. Transitioning Towards a Proactive Practice: A Longitudinal Field Study on the Implementation of a ML System in Adult Social Care. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–19.

[30] Zhechen Ren, Guang Yang, Shuoyu Wang, and Junyou Yang. 2025. Proactive care architecture for care robots. *Sci Rep* 15, 1 (Aug. 2025), 28979.

[31] Tim J Schoonbeek, Tim Houben, Hans Onvlee, Fons Van der Sommen, et al. 2024. Industreal: A dataset for procedure step recognition handling execution errors in egocentric videos in an industrial-like setting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 4365–4374. https://arxiv.org/abs/2310.17323

[32] Yasas Senarath, Ayan Mukhopadhyay, Hemant Purohit, and Abhishek Dubey. 2024. Designing a Human-centered AI Tool for Proactive Incident Detection Using Crowdsourced Data Sources to Support Emergency Response. *Digital Government: Research and Practice* 5, 1 (2024), 1–19.

[33] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. 2022. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21096–21106.

[34] Lei Shu, Nevan Wichers, Liangchen Luo, Yun Zhu, Yinxiao Liu, Jindong Chen, and Lei Meng. 2024. Fusion-Eval: Integrating Assistant Evaluators with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (Eds.). Association for Computational Linguistics, Miami, Florida, US, 225–238. doi:10.18653/v1/2024.emnlp-industry.18

[35] Denise Sogemeier, Yannick Forster, Frederik Naujoks, Josef Krems, and Andreas Keinath. 2024. The Importance of Timing—An Expert Evaluation on Latencies for Voice Assistants. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 68 (08 2024). doi:10.1177/10711813241260290

[36] Nan Sun, Bo Mao, Yongchang Li, Lumeng Ma, Di Guo, and Huaping Liu. 2024. AssistantX: An LLM-Powered Proactive Assistant in Collaborative Human-Populated Environment. arXiv:2409.17655 [cs.RO] https://arxiv.org/abs/2409.17655

[37] Miini Teng, Martin Yu, Weina Jin, Dae Woo Hwang, David Stitt, Cristina Conati, Giuseppe Carenini, Tal Jarus, and Thalia Field. 2024. Human-computer interactions and compassionate healthcare: A Wizard of Oz study using a self-administered AI-assisted cognitive assessment. *medRxiv* (2024). doi:10.1101/2024.12.17.24317999 arXiv:https://www.medrxiv.org/content/early/2024/12/20/2024.12.17.24317999.full.pdf

[38] Mrinal Verghese, Brian Chen, Hamid Eghbalzadeh, Tushar Nagarajan, and Ruta Desai. 2024. User-in-the-loop Evaluation of Multimodal LLMs for Activity Assistance. arXiv:2408.03160 [cs.CV] https://arxiv.org/abs/2408.03160

[39] 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. 2023. HoloAssist: an Egocentric Human Interaction Dataset for Interactive AI Assistants in the Real World. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 20213–20224. https://api.semanticscholar.org/CorpusID:263310718

[40] Ziqi Yang, Xuhai Xu, Bingsheng Yao, Ethan Rogers, Shao Zhang, Stephen Intille, Nawar Shara, Guodong Gordon Gao, and Dakuo Wang. 2024. Talk2care: An llm-based voice assistant for communication between healthcare providers and older adults. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 2 (2024), 1–35.

[41] Chao Zhang, Xuechen Liu, Katherine Ziska, Soobin Jeon, Chi-Lin Yu, and Ying Xu. 2024. Mathemyths: leveraging large language models to teach mathematical language through Child-AI co-creative storytelling. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–23.

[42] Jiaping Zhang, Tiancheng Zhao, and Zhou Yu. 2018. Multimodal Hierarchical Reinforcement Learning Policy for Task-Oriented Visual Dialog. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, Kazunori Komatani, Diane Litman, Kai Yu, Alex Papangelis, Lawrence Cavedon, and Mikio Nakano (Eds.). Association for Computational Linguistics, Melbourne, Australia, 140–150. doi:10.18653/v1/W18-5015

[43] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

Appendix

A Prompts

Below are the templates for all prompts used in the experiments, with example context filled in and written in blue-colored text.

A.1 Fixed Cadence Prompts

Subdialog Prompt

A human worker is performing the following physical task with the help of an AI instructor: Make coffee using a Nespresso Pixie machine

The human worker is currently on this step of the task recipe: 3.2. "Insert a Nespresso coffee capsule into the capsule compartment, with the flat side towards the front of the machine." However, the human worker only knows what to do when the AI instructor speaks the instruction aloud to them.

The instructor has spoken the full instruction for the current step to the human worker.

30.0 seconds have elapsed in this step so far.

Here is the transcript of the dialog, where the timings of each utterance are shown at the beginning of each utterance, in seconds, relative to time 0, when the instructor began delivering the instructions to the human worker:
[+0.0s] Instructor: Next, insert a Nespresso coffee capsule into the capsule compartment, with the flat side towards the front of the machine.
[+14.4s] Human Worker: what size capsule should I use
[+21.6s] Instructor: One moment please.
[+23.6s] Instructor: You can use any size of nespresso coffee capsule, as the machine is compatible with all sizes.

Based on the nature of the task, the inherent complexity of the step, the time elapsed, and the context of the dialog so far, predict whether or not the human is likely to have completed the current step by now. Remember that the human might be able to directly execute some steps quickly, while other steps might take longer for the human to comprehend the instructions, find the relevant objects, and carry out the specified action. Use your best judgement according to all evidence provided.

Please simply answer Yes or No, without any other text or explanation.

Reading Prompt

A human worker is performing the following physical task with the help of an AI instructor: Replace a hard-drive in a PC

The human worker is currently on this step of the task recipe: 1.2. "Pull the case-lid handle and open and remove the case lid. Place the lid on the table next to you." However, the human worker only knows what to do when the AI instructor speaks the instruction aloud to them.

The instructor has spoken the full instruction for the current step to the human worker.

12.0 seconds have elapsed in this step so far.

Here is the transcript of the dialog, where the timings of each utterance are shown at the beginning of each utterance, in seconds, relative to time 0, when the instructor began delivering the instructions to the human worker:
[Subdialog here as in subdialog prompt above]

The human worker has access to a visual interface that contains the text of the current instruction and a partial dialog history with the AI instructor. The human worker is currently reading the interface text. Below is their pattern of gaze behavior so far, with alternating intervals of reading the interface and gazing toward the task area. Each interval is labeled in seconds since the start of this step:
[0.0s - 6.2s]: Reading interface.
[6.2s - 11.6s]: Gazing toward task area.
[11.6s - 12.0s]: Reading interface.

Based on the nature of the task, the inherent complexity of the step, the time elapsed, and the current behavior of the human, predict whether or not the human is likely to have completed the current step by now. Remember that the human might be able to directly execute some steps quickly, while other steps might take longer for the human to comprehend the instructions, find the relevant objects, and carry out the specified action. Use your best judgement according to all evidence provided.

Please simply answer Yes or No, without any other text or explanation.

Image Prompt

A human worker is performing the following physical task with the help of an AI instructor: Make a Mojito mocktail

The human worker is currently on this step of the task recipe: 1.1: "Add 7 mint leaves in the larger tin of a Boston shaker." However, the human worker only knows what to do when the AI instructor speaks the instruction aloud to them.

The instructor is currently in the process of speaking the current instruction to the human worker, and so far has said these words: "First, add 7 mint...."

4.0 seconds have elapsed in this step so far.

Here is the transcript of the dialog, where the timings of each utterance are shown at the beginning of each utterance, in seconds, relative to time 0, when the instructor began delivering the instructions to the human worker:
[Subdialog here as in subdialog prompt above]

This image shows the human's point-of-view at this exact moment (note that it may be blurry or occluded):

Based on the nature of the task, the inherent complexity of the step, the time elapsed, and the context of the visual scene at this moment, predict whether or not the human is likely to have completed the current step by now. Remember that the human might be able to directly execute some steps quickly, while other steps might take longer for the human to comprehend the instructions, find the relevant objects, and carry out the specified action. The provided image may be blurry or occluded, but use your best judgement according to all evidence provided.

Please simply answer Yes or No, without any other text or explanation.

All (Subdialog + Reading + Image Prompt)
A human worker is performing the following physical task with the help of an AI instructor: Make coffee using a Nespresso Pixie machine

The human worker is currently on this step of the task recipe: 1.2. "Fill the water tank with fresh water" However, the human worker only knows what to do when the AI instructor speaks the instruction aloud to them.

The instructor has spoken the full instruction for the current step to the human worker.

8.0 seconds have elapsed in this step so far.

Here is the transcript of the dialog, where the timings of each utterance are shown at the beginning of each utterance, in seconds, relative to time 0, when the instructor began delivering the instructions to the human worker:
[+0.0s] Instructor: Next, fill the water tank with fresh water.
[+7.3s] Human Worker: how much ...
The human worker is currently in the process of speaking their utterance.

The human worker has access to a visual interface that contains the text of the current instruction and a partial dialog history with the AI instructor. The human worker is currently reading the interface text. Below is their pattern of gaze behavior so far, with alternating intervals of reading the interface and gazing toward the task area. Each interval is labeled in seconds since the start of this step:
[0.0s - 0.2s]: Reading interface.
[0.2s - 1.2s]: Gazing toward task area.
[1.2s - 1.8s]: Reading interface.
[1.8s - 7.6s]: Gazing toward task area.
[7.6s - 8.0s]: Reading interface.

This image shows the human's point-of-view at this exact moment (note that it may be blurry or occluded):

Based on the nature of the task, the inherent complexity of the step, the time elapsed, and and all other context provided, predict whether or not the human is likely to have completed the current step by now. Remember that the human might be able to directly execute some steps quickly, while other steps might take longer for the human to comprehend the instructions, find the relevant objects, and carry out the specified action. The provided image may be blurry or occluded, but use your best judgement according to all evidence provided.

Please simply answer Yes or No, without any other text or explanation.

Video Prompt
A human worker is performing the following physical task with the help of an AI instructor: Make a Mojito mocktail

The human worker is currently on this step of the task recipe: 1.1: "Add 7 mint leaves in the larger tin of a Boston shaker." However, the human worker only knows what to do when the AI instructor speaks the instruction aloud to them.

The instructor is currently in the process of speaking the current instruction to the human worker, and so far has said these words: "First, add 7 mint...."

4.0 seconds have elapsed in this step so far.

Here is the transcript of the dialog, where the timings of each utterance are shown at the beginning of each utterance, in seconds, relative to time 0, when the instructor began delivering the instructions to the human worker:
[Subdialog here as in subdialog prompt above]

This video shows the human's point-of-view from the beginning of the step (or 10 seconds ago if more than 10 seconds have elapsed) until now: *[POV video clip here]*
Based on the nature of the task, the inherent complexity of the step, the time elapsed, and the context including the POV visual scene so far, predict whether or not the human is likely to have completed the current step by now. Remember that the human might be able to directly execute some steps quickly, while other steps might take

longer for the human to comprehend the instructions, find the relevant objects, and carry out the specified action. The provided video may sometimes be blurry or occluded, but use your best judgement according to all evidence provided.

Please simply answer Yes or No, without any other text or explanation.

Image vs Expert Reference Image Prompt
A human worker is performing the following physical task with the help of an AI instructor: Make a whiskey sour mocktail

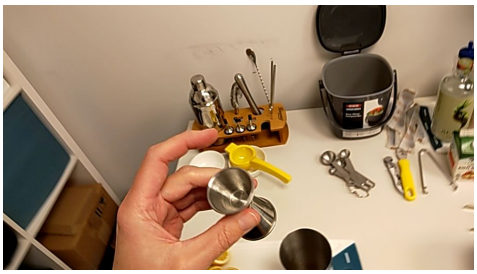
The human worker is currently on this step of the task recipe: 1.4. "Pour 3/4 ounces of the freshly squeezed lemon juice into the same tin." However, the human worker only knows what to do when the AI instructor speaks the instruction aloud to them.

The instructor has spoken the full instruction for the current step to the human worker.

12.0 seconds have elapsed in this step so far.

Here is the transcript of the dialog, where the timings of each utterance are shown at the beginning of each utterance, in seconds, relative to time 0, when the instructor began delivering the instructions to the human worker:
[Subdialog here as in subdialog prompt above]

This image shows the human's point-of-view at this exact moment (note that it may be blurry or occluded):



Here's how the scene looked like when a totally different human, with high knowledge and high task expertise, successfully completed the step in 12.0 seconds:

Based on the nature of the task, the inherent complexity of the step, the time elapsed, and in comparison to what the scene should look like when an expert executes the step quickly and correctly, predict whether or not this human is likely to have completed the current step by now. Remember that this human might be able to directly execute some steps quickly, while other steps might take longer for the human to comprehend the instructions, find the relevant objects, and carry out the specified action.

Please simply answer Yes or No, without any other text or explanation.

A.2 One-Time Prompt

A human worker is performing the following physical task with the help of an AI instructor: Make a notebook using a binding machine

The human worker is currently on this step of the task recipe: 2.1. "Take a front and back cover and place them in the binding machine aligning with the edge guide."

The human worker only knows what to do when the AI instructor speaks the instruction aloud to them, and the AI instructor is just beginning to speak now. It will take 6.1 seconds for it to finish speaking the instruction.

Based on the nature of the task and the inherent complexity of the step, predict how long you think it will take the human worker to complete this step, starting from now. Remember that the human worker might be able to directly execute some steps quickly, even in parallel with the instructor's speech, while other steps might take longer for the human worker to comprehend the instructions, find the relevant objects, and carry out the specified action.

Predict how long it will take the human worker to correctly execute this step, starting from now. Use your best judgement according to all evidence provided, and provide your best guess estimate including seconds and hundreds of milliseconds using the json format below:

```
{
"totalseconds": "x.x",
}
```

Provide your response exactly matching the JSON format described above and make sure you provide a well-formatted JSON. Do not provide any text before or after the JSON, just start with an open brace and end with a closing brace.

A.3 Dynamic Prompt

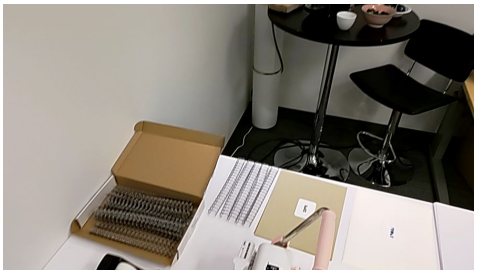
Initial prompt at beginning of sub-step:
A human worker is performing the following physical task with the help of an AI instructor: Make a notebook using a binding machine

The human worker is currently on this step of the task recipe: 2.1. "Take a front and back cover and place them in the binding machine aligning with the edge guide."

The human worker only knows what to do when the AI instructor speaks the instruction aloud to them, and the AI instructor is just

beginning to speak now. It will take 6.1 seconds for it to finish speaking the instruction.

This image shows the human's point-of-view at this exact moment (note that it may be blurry or occluded):



Based on the nature of the task, the inherent complexity of the step, and the context of the visual scene at this moment, predict how long you think it will take the human to complete this step, starting from now. Remember that the human might be able to directly execute some steps quickly, even in parallel with the instructor's speech, while other steps might take longer for the human to comprehend the instructions, find the relevant objects, and carry out the specified action.

Predict how long it will take the human to correctly execute this step, starting from now. If you believe the human has *already* finished this step from the very beginning, provide an estimate of 0 or 0.0 total seconds. Use your best judgement according to all evidence provided, and provide your best guess estimate including seconds and hundreds of milliseconds using the json format below:

{
"totalseconds": "x.x",
}

Provide your response exactly matching the JSON format described above and make sure you provide a well-formatted JSON. Do not provide any text before or after the JSON, just start with an open brace and end with a closing brace.

Subsequent prompts throughout sub-step:
A human worker is performing the following physical task with the help of an AI instructor: Make a notebook using a binding machine

The human worker is currently on this step of the task recipe: 2.1. "Take a front and back cover and place them in the binding machine aligning with the edge guide."

The human only knows what to do when the AI assistant speaks the instruction aloud to them.

The instructor has spoken the full instruction for the current step to the human worker.

8.0 seconds have elapsed in this step so far.

Here is the transcript of the dialog, where the timings of each utterance are shown at the beginning of each utterance, in seconds, relative to time 0, when the instructor began delivering the instructions to the human worker:
[+0.0s] Instructor: First, take a front and back cover and place them in the binding machine aligning with the edge guide.

The human worker has access to a visual interface that contains the text of the current instruction and a partial dialog history with the AI instructor. The human worker is not currently reading the interface text. They are currently gazing toward the task area. Below is their pattern of gaze behavior so far, with alternating intervals of reading the interface and gazing toward the task area. Each interval is labeled in seconds since the start of this step:
[0.0s - 0.5s]: Reading interface.
[0.5s - 3.8s]: Gazing toward task area.
[3.8s - 5.2s]: Reading interface.
[5.2s - 8.0s]: Gazing toward task area.

This image shows the human's point-of-view at this exact moment (note that it may be blurry or occluded):

Based on the nature of the task, the inherent complexity of the step, the time elapsed, and all other context provided, predict whether or not the human is likely to have completed the current step by now. Your previous prediction was that the human would need 8.0 seconds to complete the step, and that time has now elapsed.

Remember that the human might be able to directly execute some steps quickly, even in parallel with the instructor's speech, while other steps might take longer for the human to comprehend the instructions, find the relevant objects, and carry out the specified action.

Please answer Yes or No in the JSON structure below. If your answer is No, predict how much additional time you think it will take the human to complete the step, in seconds and tenths of seconds.

{
"stepcomplete": "Yes" or "No",
"remainingseconds": "x.x",
}

Provide your response exactly matching the JSON format described above and make sure you provide a well-formatted JSON. Do not provide any text before or after the JSON, just start with an open brace and end with a closing brace.

B All Results

Table 2: Performance across all prompt versions, models, and baselines.

Prompt Version	Model	Accuracy (Query)	F1 (Query)	Accuracy (Step)	F1 (Step)	False Positive Percentage (Step)	Percent Optimal Time Saved
Baseline(0)	N/A	N/A	N/A	36.5	0.518	39.6%	35.8%
Baseline(+1s)	N/A	N/A	N/A	31.0	0.451	34.6%	33.0%
Baseline(+2s)	N/A	N/A	N/A	27.8	0.408	31.3%	30.5%
Baseline(+3s)	N/A	N/A	N/A	25.0	0.370	28.6%	30.2%
Baseline(+4s)	N/A	N/A	N/A	22.6	0.337	27.1%	28.2%
Baseline(+5s)	N/A	N/A	N/A	21.1	0.311	25.4%	27.2%
Baseline(+10s)	N/A	N/A	N/A	16.7	0.232	18.7%	23.6%
Baseline(+15s)	N/A	N/A	N/A	14.1	0.178	14.8%	20.5%
Baseline(+20s)	N/A	N/A	N/A	12.6	0.143	11.8%	19.7%
Subdialog	Llama-3.2	71.7	0.263	23.4	0.361	57.3%	19.3%
	Qwen-3.5	69.1	0.406	26.7	0.404	63.3%	21.6%
	GPT-5.1	76.3	0.441	29.5	0.427	41.7%	24.6%
	GPT-5.4	71.5	0.441	28.2	0.416	53.0%	23.0%
	Gemini-3.1-Pro	75.0	0.446	33.3	0.477	39.9%	24.9%
Reading	Llama-3.2	74.8	0.085	15.5	0.198	27.0%	10.4%
	Qwen-3.5	71.2	0.358	24.9	0.374	54.3%	22.3%
	GPT-5.1	76.0	0.381	27.1	0.387	32.7%	26.1%
	GPT-5.4	71.8	0.376	25.1	0.372	46.5%	24.5%
	Gemini-3.1-Pro	76.3	0.365	33.0	0.467	28.8%	26.8%
Image	Llama-3.2	50.1	0.341	5.9	0.111	94.1%	14.4%
	Qwen-3.5	76.3	0.365	28.3	0.416	50.0%	26.3%
	GPT-5.1	77.0	0.363	27.8	0.396	37.4%	32.6%
	GPT-5.4	78.0	0.312	32.8	0.456	20.4%	32.3%
	Gemini-3.1-Pro	81.3	0.544	45.0	0.604	35.8%	40.1%
All	Llama-3.2	54.3	0.313	7.8	0.145	91.3%	15.2%
	Qwen-3.5	75.8	0.333	26.8	0.394	46.9%	26.3%
	GPT-5.1	75.6	0.355	25.2	0.368	44.3%	29.7%
	GPT-5.4	77.3	0.267	29.0	0.407	22.3%	26.0%
	Gemini-3.1-Pro	81.6	0.554	48.2	0.636	35.7%	44.9%

Continued on next page

Table 2 – continued from previous page

Prompt Version	Model	Accuracy (Query)	F1 (Query)	Accuracy (Step)	F1 (Step)	False Positive Percentage (Step)	Percent Optimal Time Saved
Reference	Qwen-3.5	71.0	0.359	19.6	0.312	69.0%	17.3%
	GPT-5.1	71.1	0.449	22.7	0.342	56.1%	21.1%
	GPT-5.4	77.2	0.361	31.0	0.438	27.9%	26.2%
	Gemini-3.1-Pro	78.4	0.493	38.8	0.542	39.8%	34.0%
Video	Gemini-3.1-Pro	82.6	0.655	50.0	0.655	47.9%	45.3%
One-Time	Llama-3.2	N/A	N/A	23.4	0.371	68.0%	24.4%
	Qwen-3.5	N/A	N/A	28.2	0.426	57.4%	23.9%
	GPT-5.1	N/A	N/A	29.3	0.436	51.7%	27.0%
	GPT-5.4	N/A	N/A	31.1	0.466	58.1%	26.1%
	Gemini-3.1-Pro	N/A	N/A	29.7	0.443	51.7%	25.6%
Dynamic	Llama-3.2	N/A	N/A	8.7	0.024	3.0%	0.7%
	Qwen-3.5	N/A	N/A	23.0	0.322	25.4%	23.5%
	GPT-5.1	N/A	N/A	26.7	0.398	48.8%	27.7%
	GPT-5.4	N/A	N/A	28.0	0.398	32.3%	26.9%
	Gemini-3.1-Pro	N/A	N/A	39.3	0.530	11.0%	31.9%