

Pro²Assist: Continuous Step-aware Proactive Assistance with Multi-modal Egocentric Perception for Long-horizon Procedural Tasks

LILIN XU[†], Columbia University, United States

BUFANG YANG[†], The Chinese University of Hong Kong, China

SIYANG JIANG, The Chinese University of Hong Kong, China

KAIWEI LIU, The Chinese University of Hong Kong, China

KAIYUAN HOU, Columbia University, United States

YUANG FAN, Columbia University, United States

HONGKAI CHEN, The Chinese University of Hong Kong, China

ZHENYU YAN, The Chinese University of Hong Kong, China

XIAOFAN JIANG*, Columbia University, United States

Procedural tasks with multiple ordered steps are ubiquitous in daily life. Recent advances in multimodal large language models (MLLMs) have enabled personal assistants that support daily activities. However, existing systems primarily provide reactive guidance triggered by user queries, or limited proactive assistance for isolated short-term events rather than long-horizon procedural tasks. In this work, we introduce Pro²Assist, a step-aware proactive assistant that continuously tracks fine-grained task progress and reasons over the user's evolving state to provide timely assistance throughout tasks. Pro²Assist leverages multimodal data from augmented reality (AR) glasses to achieve motion-based perception. It then extracts step-oriented procedural context from multi-scale temporal dynamics and task-specific expert knowledge. Based on both sensory input and procedural context, Pro²Assist performs continuous reasoning to infer user needs and display timely assistance on AR glasses. We evaluate Pro²Assist using a dataset curated from public sources and a real-world dataset collected on our testbed with AR glasses. Extensive evaluations show that Pro²Assist outperforms the best-performing baselines by over 21% in procedural action understanding accuracy, and it achieves up to 2.29× the proactive timing accuracy of baselines. A user study with 20 participants further shows that 90% find Pro²Assist useful, indicating its effectiveness for real-world procedural assistance.

CCS Concepts: • **Human-centered computing** → Ubiquitous and mobile computing; • **Computing methodologies** → Artificial intelligence.

Additional Key Words and Phrases: Vision-Language Models, Multi-modal Egocentric Data, AR Glasses, Wearable Sensing, Proactive Assistive Systems, Internet of Things

[†]Co-author equal contribution.

*Corresponding author.

Authors' Contact Information: [Lilin Xu](#), Columbia University, New York, United States, lilin.xu@columbia.edu; [Bufang Yang](#), The Chinese University of Hong Kong, Hong Kong, China, bfiyang@link.cuhk.edu.hk; [Siyang Jiang](#), The Chinese University of Hong Kong, Hong Kong, China, syjiang@ie.cuhk.edu.hk; [Kaiwei Liu](#), The Chinese University of Hong Kong, Hong Kong, China, 1155189693@link.cuhk.edu.hk; [Kaiyuan Hou](#), Columbia University, New York, United States, kh3119@columbia.edu; [Yuang Fan](#), Columbia University, New York, United States, yf2676@columbia.edu; [Hongkai Chen](#), The Chinese University of Hong Kong, Hong Kong, China, hkchen@ie.cuhk.edu.hk; [Zhenyu Yan](#), The Chinese University of Hong Kong, Hong Kong, China, zyyan@ie.cuhk.edu.hk; [Xiaofan Jiang](#) (corresponding author), Columbia University, New York, United States, jiang@ee.columbia.edu.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2026 Copyright held by the owner/author(s).

ACM 2474-9567/2026/9-ART178

<https://doi.org/10.1145/3832022>

ACM Reference Format:

Lilin Xu, Bufang Yang, Siyang Jiang, Kaiwei Liu, Kaiyuan Hou, Yuang Fan, Hongkai Chen, Zhenyu Yan, and Xiaofan Jiang. 2026. Pro²Assist: Continuous Step-aware Proactive Assistance with Multi-modal Egocentric Perception for Long-horizon Procedural Tasks. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 10, 3, Article 178 (September 2026), 31 pages. <https://doi.org/10.1145/3832022>

1 INTRODUCTION

Procedural tasks are ubiquitous in daily life and play a crucial role in many routine human activities, spanning from cooking to assembling everyday items [10, 40]. These tasks typically involve multiple steps that need to be executed in precise order, which can be challenging when the procedure is complex or unfamiliar to the user. Although there are usually instruction manuals and online tutorials available, they require users to repeatedly shift attention between physical actions and external references, leading to cognitive interruptions and increased mental load [52, 56]. With the rapid advancement of LLMs and MLLMs [4, 13, 55], intelligent personal assistants have been developed to handle users' questions by providing relevant task instructions [10, 35], thereby reducing the need for manual instruction lookup during procedural tasks.

However, most of these procedural assistants are reactive, requiring explicit user queries that interrupt ongoing actions and undermine seamless task guidance. Compared to reactive assistants, recent research has proposed proactive systems [45, 73, 74] that aim to further reduce users' physical and mental workload by inferring when and what assistance to provide without waiting for explicit queries. These systems can recognize short-term events and proactively assist with them, such as detecting a user viewing products and offering price comparisons. However, most existing proactive systems provide one-shot¹ assistance at the level of an isolated event based on holistic scene understanding, rather than continuous, step-by-step guidance. As a result, they are less suitable for long-horizon procedural tasks with multiple steps, where user needs evolve over time and are strongly correlated with task progress, as illustrated on the right side of Figure 1. Although recent works [11, 12, 41] explore proactive interventions for procedural tasks, they rely on discrete trigger events and primarily deliver content about the next step. These gaps highlight the need for *continuous, step-aware* assistants that understand the user's ongoing actions and deliver timely guidance grounded in the user's actual state throughout long-horizon procedural tasks.

To address this gap, we aim to develop a procedural assistive system that continuously tracks fine-grained task progress, including the current procedural step and within-step execution state (e.g., just started or about to finish), and reasons over the user's evolving state to help users perform tasks smoothly. However, developing such a system introduces several unique challenges. *First*, procedural tasks involve continuous interactions between the user and the physical environment, where the user's attention and state are crucial for identifying assistive moments. However, existing works [20, 67] mainly rely on egocentric vision and often overlook implicit attention cues from head and hand motion, which leads to limited intent and attention understanding, weakening the timeliness of assistance. *Second*, continuously tracking fine-grained task progress requires capturing temporal dynamics, including short-term hand manipulations and long-term historical context, together with procedural knowledge, rather than relying on isolated single-moment observations. However, prior work [20] mainly relies on vision-only dense frame sequences to capture temporal context without explicitly modeling user intent or procedural knowledge, posing challenges for correctly interpreting fine-grained task progress. *Third*, while existing VLMs [59, 76, 79] are developed for general scene and action understanding, they exhibit insufficient capability in understanding users' ongoing actions in long-horizon and temporally dependent procedural tasks. This makes it difficult to continuously reason over the user's evolving state, limiting the ability to deliver assistance grounded in the user's actual state.

¹Throughout this paper, "one-shot" refers to providing isolated assistance for the overall event based on holistic scene understanding

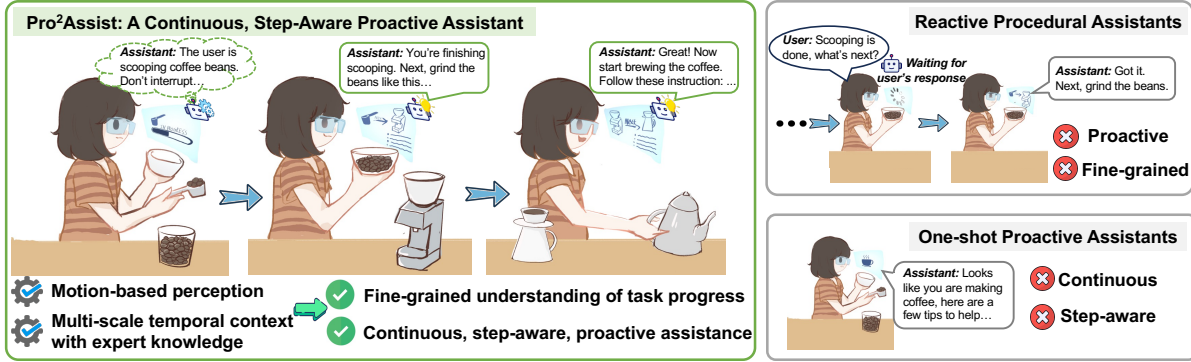


Fig. 1. Application scenario of Pro²Assist. Pro²Assist provides continuous, step-aware proactive assistance by reasoning over the user's evolving state in the task workflow (left), while reactive procedural assistants rely on explicit user requests and one-shot proactive assistants provide isolated assistance for the overall event (right).

In this paper, we introduce Pro²Assist, a continuous, step-aware Proactive Assistant for long-horizon Procedural tasks that integrates multimodal egocentric perception and LLM reasoning to deliver timely assistance through AR glasses, as shown in Figure 1's left side. Pro²Assist first introduces an motion-based perception mechanism based on multimodal egocentric data from AR glasses. It integrates head-motion-aware sampling with key moment selection based on optical flow estimation over visual data to identify moments with high potential for assistive needs from continuous observations. Next, Pro²Assist performs step-oriented procedural context extraction by incorporating task-specific expert knowledge together with multi-scale temporal context, including short-term hand motion cues and long-term task progress. Finally, combining the extracted procedural context with sensory context, Pro²Assist performs step-aware proactive reasoning to infer users' current states and needs, enabling continuous assistance aligned with fine-grained task progress. To ensure reliable and non-intrusive assistance under continuous reasoning, Pro²Assist also introduces a step-aware consistency checking mechanism that utilizes historical predictions to mitigate single-moment errors and improve response timing.

We implement Pro²Assist on a real-world testbed with AR glasses and a back-end server. We conduct extensive evaluations on both a dataset curated from three public datasets and a real-world dataset collected on our testbed. Results show that Pro²Assist achieves effective proactive step-aware assistance, significantly outperforming state-of-the-art baselines. Compared with the best-performing baselines, Pro²Assist improves step identification accuracy by 25.2%, execution status identification accuracy by 21.6%, and proactive accuracy by 15.1%, and it achieves up to 2.29× the proactive timing accuracy of the baselines. The results also show that Pro²Assist is robust across different VLM scales and system settings. In addition, a user study with 20 participants indicates that 90% found Pro²Assist useful, with particularly strong agreement among users unfamiliar with the tasks, highlighting its practical effectiveness. We summarize the contributions of this paper as follows.

- We present Pro²Assist, an end-to-end assistive system that delivers continuous, step-aware assistance throughout procedural tasks by observing and reasoning over multimodal egocentric data from AR smart glasses.
- We propose a motion-based perception mechanism and a step-oriented procedural context extraction approach for efficient continuous perception and fine-grained task progress tracking. Pro²Assist first utilizes attention cues by combining head motion signals with visual motion cues to identify moments that are likely to require proactive assistance. It then effectively extracts procedural context by integrating multi-scale temporal context with task-specific expert knowledge.

- We develop a step-aware proactive reasoner that reasons over the user's evolving task state rather than isolated observations, with a consistency checking mechanism that leverages temporal consistency in continuous reasoning, enabling timely and non-intrusive assistance aligned with fine-grained task progress.
- We implement Pro²Assist on a real-world testbed with AR glasses and a back-end server. Extensive evaluations on both a curated dataset and a real-world dataset demonstrate that Pro²Assist significantly outperforms state-of-the-art baselines in procedural action understanding and proactive assistance performance, highlighting its effectiveness in delivering timely and step-aware proactive assistance. A user study shows 90% of participants find Pro²Assist useful, further indicating its practical effectiveness.

2 RELATED WORK

2.1 Egocentric Smart Assistants

Egocentric perception has been widely adopted for applications such as object recognition [9, 66], action recognition [39], and error detection [17, 40]. Beyond these task-specific applications, recent studies have developed egocentric smart assistants that leverage large language and vision models to support everyday human activities [35, 68, 70]. MM-Ego [76] and EgoLife [75] collect large-scale egocentric vision data paired with text and develop foundation models for egocentric question answering and personal assistance. Several studies [16, 23, 29, 69, 79] also develop diverse datasets and benchmarks for egocentric video assistants. Beyond foundation models and datasets, recent work has developed real-world systems that integrate diverse sensor data from mobile and wearable platforms, such as smart glasses, for everyday use. Vinci [35] leverages egocentric video to provide real-time responses to user queries based on its observations and historical context. agentAR [80] develops a personal agent with an AR authoring system that integrates external tools and LLM-based reasoning to support personal question-answering tasks. OS-1 [68] is a companion on smart glasses that uses visual and audio cues from the environment to deliver personalized responses. Several studies have explored egocentric assistants for broader applications, such as assisting individuals with visual impairments [58, 72] and supporting for social interaction [62, 78]. While these studies primarily focus on question-answering scenarios that require explicit user instructions, Pro²Assist aims to proactively provide assistance throughout procedural tasks based on multimodal egocentric sensor data from smart glasses.

2.2 Continual and Procedural Personal Assistants

Recent studies, such as Gemini [4], utilize understanding capabilities of MLLMs in live visual perception to serve as personal assistants that provide real-time daily support. However, these systems require users to explicitly specify their task progress or current step to trigger guidance for subsequent steps. PrISM-Q&A [10] is a voice assistant that uses audio and motion sensors for human activity recognition (HAR) and provides step-aware support for procedural tasks. However, it remains constrained to a question-answering paradigm, requiring users to explicitly ask for guidance during procedural tasks. Building on HAR-based step tracking, PrISM-Observer [12] enables proactive intervention by predicting intervention moments for a set of pre-selected steps through modeling step durations and transitions. The PrISM framework [11] further addresses HAR's sensing imperfections through extracting step context from continuous mixed-initiative dialogue (e.g., reactive Q&A, user self-narration, and system reminders). Satori [41] forecasts next-step assistance in parallel while detecting step-completion checkpoints to trigger delivery of the cached assistance through user confirmation. These systems trigger assistance based on discrete events (e.g., predicted time-to-step, step completion detection) and primarily deliver pre-selected or pre-forecast content for the next step. In contrast, Pro²Assist exploits multimodal data from smartglasses to continuously track fine-grained task progress and reason over the user's evolving state, deciding at each moment whether and what assistance to deliver based on the user's actual state.

Table 1. A summary of recent egocentric assistive systems (● means included. ‘V’ and ‘A’ denote vision and audio).

Approach	Adaptive Perception	Procedural Tasks	Step-Aware Assistance	Expert Knowledge	Sensor Modalities	Interactive Mode	Assistance Mode	System Settings
MM-Ego [76]	○	○	○	○	V	Reactive	One-shot	N.A.
Vinci [35]	○	●	●	○	V	Reactive	Continuous	Glasses
agentAR [80]	○	●	●	○	V	Reactive	Continuous	Glasses
PrISM-Q&A [10]	○	●	●	●	A, IMU	Reactive	Continuous	Smartwatch
PrISM-Observer [12]	○	●	●	○	A, IMU	Proactive	Continuous	Smartwatch
PrISM [11]	○	●	●	●	A, IMU	Mixed	Continuous	Smartwatch
VideoLLM-online [20]	○	○	○	○	V	Proactive	Continuous	N.A.
ContextAgent [74]	○	○	○	○	V	Proactive	One-shot	N.A.
ProAgent [73]	●	○	○	○	V, A, IMU, GPS	Proactive	One-shot	Glasses
SocialMind [70]	○	○	○	○	V, A, IMU	Proactive	Continuous	Glasses
OS-1 [68]	○	○	○	○	V, A	Proactive	One-shot	Glasses
Pro²Assist	●	●	●	●	V, A, IMU	Proactive	Continuous	Glasses

2.3 Proactive Assistant Systems

Recent studies, such as VideoLLM-online [20], explore enabling VLMs to support online interaction over streaming video by introducing training objectives that allow the model to determine when to respond while processing dense video sequences. However, this line of work focuses on efficient processing of visual streams at the VLM level and does not explicitly model user intent, which is essential for providing appropriate assistance. Beyond model-level designs, recent research has proposed proactive systems that model user intent to anticipate user needs and provide assistance without explicit user instructions [45, 51, 73, 74]. Studies such as SocialMind [70] and LLAMAPIE [21] provide proactive suggestions on AR glasses or earphones during face-to-face conversations, with a primary focus on social scenarios. ContextAgent [74], ProAgent [73], and ChainStream [45] can harness multimodal sensor data for reasoning and automatically decide when and what to proactively assist users. ProMemAssist [51] focuses on modeling the assistance value and interruption cost, enabling more selective proactive support during ongoing tasks. However, these studies mainly focus on delivering one-shot proactive support for short-term events and isolated moments. In contrast, Pro²Assist provides continuous, step-aware assistance by incorporating multi-scale temporal procedural context and expert knowledge into reasoning, rather than relying on single-moment holistic scene understanding.

3 BACKGROUND AND MOTIVATION

In this section, we present observations and measurements from self-collected egocentric recordings to motivate the design of Pro²Assist, using the procedural task of tea making as an example.

Observation 1: Continuous, Step-Aware Assistance in Procedural Tasks. Unlike general scenarios where one-shot suggestions for isolated short-term events are sufficient [73, 74], procedural tasks consist of ordered, interdependent steps whose assistance should continuously reason over the user’s current step and execution status throughout the task. As shown in Figure 2, when the user is about to finish checking the water temperature during tea making, the one-shot proactive system correctly recognizes the kitchen scene but fails to deliver appropriate assistance due to inaccurate step and status recognition. Effective proactive assistance in procedural tasks should provide step-specific instructions at the beginning of an action and suggest the next step as the current action nears completion. Therefore, continuous, step-aware assistance is essential for procedural tasks, motivating a design that continuously reasons over the user’s evolving state across interdependent steps to deliver timely guidance grounded in the user’s actual state rather than one-shot suggestions.

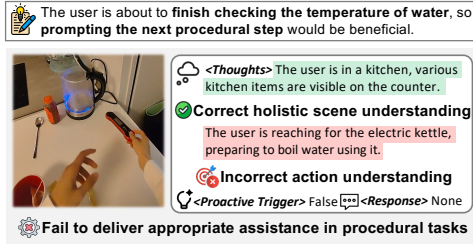


Fig. 2. An example of an existing proactive system in procedural tasks [73].

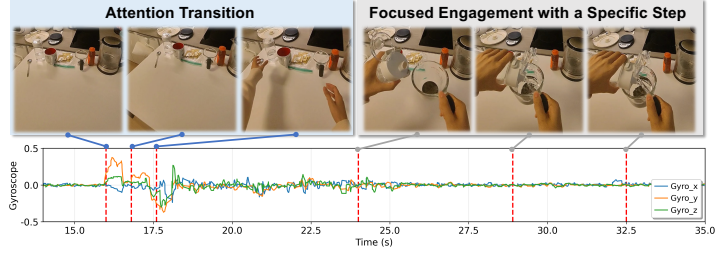


Fig. 3. A preliminary example that head motion can effectively indicate attention transitions during procedural task execution.

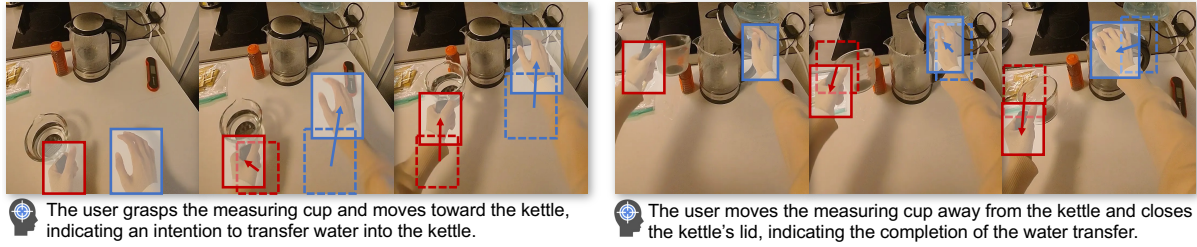


Fig. 4. Egocentric vision and hand motion provide important cues for inferring user intent during procedural tasks.

Observation 2: User's Intent and Attention in Procedural Task Execution. Proactive assistance requires an accurate understanding of user intent. In particular, correctly identifying the user's attentional focus is essential for providing timely and relevant assistance in procedural tasks. We observe that head and hand motions captured by AR glasses serve as key indicators. On the one hand, head motion is a strong indicator of attention transitions [26, 27]. As shown in Figure 3, when the user transitions to the step of measuring cold water and searches for the water bottle while making tea, head motion exhibits significant movement patterns. In contrast, when users focus on a specific manipulation, like pouring water into the measuring cup, head motion remains stable. On the other hand, hand manipulation in egocentric vision is critical for inferring user intent and execution status [14, 15, 40]. As shown in Figure 4, hand manipulation directly reflects the user's focus and step execution status, providing rich temporal information about short-term procedural dynamics. Therefore, head and hand motions serve as cues of user intent and attention in procedural tasks, motivating our use of egocentric multimodal data from AR glasses to capture these signals.

Observation 3: Action Understanding for Procedural Tasks. As shown in Figure 5, action understanding in procedural tasks exhibits two characteristics. First, visually similar but functionally opposite manipulations within the same step, such as "tilting a cup to pour water" versus "lifting the cup after pouring" in Figure 5 (a), should be distinguished to identify execution status. Distinguishing them benefits from short-term hand manipulation cues that capture temporal dynamics within a step. Second, similar actions belonging to different steps, such as "placing a tea bag into a mug" versus "steeping the tea bag" in Figure 5 (b), share similar visual appearance but correspond to different procedural stages. Distinguishing them benefits from both task-specific expert knowledge, which provides step ordering, dependencies, and step-level execution details, and long-term task progress, which tracks completed steps throughout task execution. Together, these two sources offer complementary information that helps better understand the user's ongoing action. For example, knowing that "steeping" must follow "placing the tea bag" and that "placing" has already been completed allows the system to correctly identify the current action as "steeping". We adapt existing VLMs of varying scales to procedural action understanding



Fig. 5. Examples illustrating two characteristics in action understanding for procedural tasks: visually similar but functionally inverse manipulations within the same step (left), and similar actions that correspond to different steps (right).

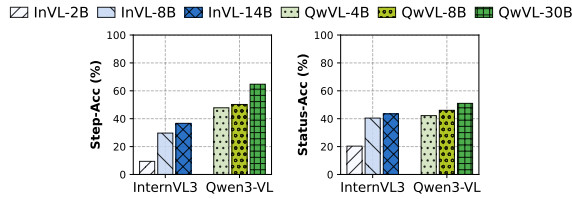


Fig. 6. Performance of existing VLMs adapted for procedural action understanding. “Step-Acc” and “Status-Acc” denote step and execution status identification accuracy, respectively.

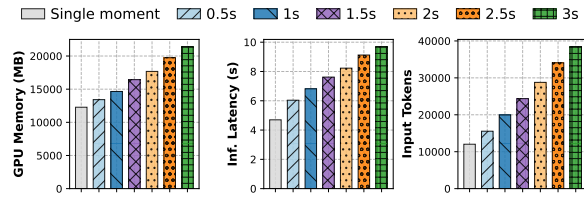


Fig. 7. System overhead increases significantly as more input frames are incorporated as temporal context. “Inf.” denotes inference.

using in-context learning [25] with five examples. As shown in Figure 6, existing VLMs struggle to identify both the step and its execution status when relying solely on intrinsic knowledge and single-moment reasoning, as procedural tasks are inherently sequential and interdependent. Meanwhile, as shown in Figure 7, incorporating temporal context directly through dense frame sequences incurs rapidly growing computational overhead, posing challenges for real-time assistance. Therefore, both short-term and long-term temporal context and task-specific expert knowledge are essential for effective procedural action understanding, motivating a design that integrates procedural knowledge with efficient temporal modeling for real-time assistance.

4 SYSTEM DESIGN

4.1 System Overview

Pro²Assist is an end-to-end assistive system that exploits multimodal egocentric sensor data from AR glasses to provide continuous, step-aware assistance during procedural tasks. Figure 8 demonstrates the overview of Pro²Assist, which consists of three major modules. First, Pro²Assist performs *motion-based perception* (§ 4.2) by adaptively sampling visual data using a head-motion-aware strategy and extracting key moments via motion-based selection. This design utilizes attention cues from multimodal egocentric data to reduce redundant visual processing while preserving moments with high potential for proactive needs. Second, Pro²Assist achieves *step-oriented procedural context extraction* (§ 4.3) by effectively incorporating task-specific expert knowledge about the particular procedural task and multi-scale temporal context, including short-term hand motion cues and long-term task progress. The extracted context captures both procedural knowledge and temporal dynamics, which are important for procedural action understanding. Finally, *step-aware proactive reasoner* (§ 4.4) in Pro²Assist integrates sensory context with the extracted procedural context to perform motion-aware action understanding and step-aware proactive reasoning. Proactive responses are generated only when necessary, and a step-aware consistency checking mechanism is introduced to suppress redundant feedback and mitigate single-moment mispredictions by leveraging temporal consistency across historical predictions. These responses are then displayed on the AR glasses, allowing users to keep the task scene in view while receiving timely assistance.

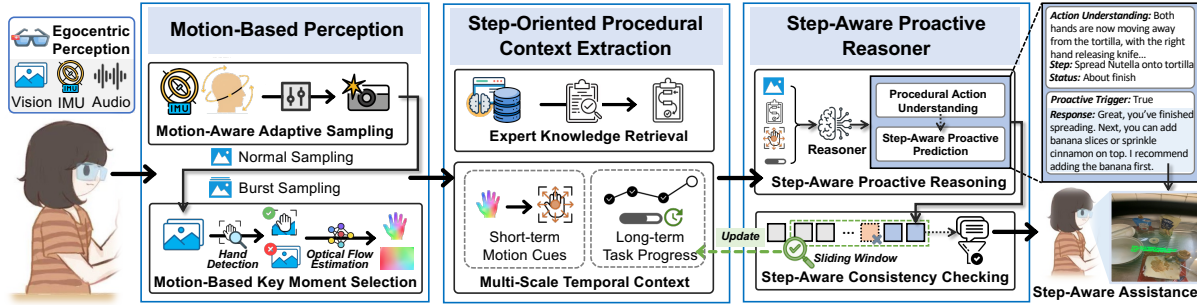


Fig. 8. System overview of Pro²Assist. Pro²Assist utilizes multimodal egocentric data from AR glasses to achieve motion-based perception. By integrating visual inputs with extracted task-specific expert knowledge and multi-scale temporal context, the reasoner performs step-aware proactive reasoning with consistency checking. The resulting proactive assistance is then delivered to the user via on-screen displays on the AR glasses.

4.2 Motion-Based Perception

Unlike reactive assistants [35, 68] triggered by explicit queries, a proactive assistant must continuously observe user state to identify moments requiring assistance. However, such moments are typically sparse in long egocentric streams, making uniform processing with intent inference inefficient and unnecessary. As discussed in §3 (Observation 2), motion cues from multimodal egocentric sensing can indicate user intent and attention. Thus, Pro²Assist introduces a motion-based perception mechanism for efficient intent inference, leveraging always-on, low-cost head motion signals to adaptively guide visual sampling and fine-grained visual motion cues to further identify moments with high potential for proactive needs from continuous observations.

4.2.1 Head-Motion-Aware Adaptive Sampling. To efficiently decide how densely to capture frames, Pro²Assist exploits head motion from IMU as an always-on sampling signal to drive adaptive vision sampling, as it is far cheaper than continuous vision processing, while effectively reflecting potential attention transitions during procedural tasks. Specifically, Pro²Assist measures head motion as the magnitude of the angular velocity measured by the gyroscope, which captures rotational head movements with low cost and latency. Pro²Assist continuously monitors head motion and applies a motion-aware sampling threshold τ_s to determine the visual data sampling rate. When head motion remains below τ_s , indicating stable head orientation, the system operates in a normal sampling mode, which still periodically captures visual observations of ongoing task execution, including moments when users focus on specific manipulations within a step. Once head motion exceeds τ_s , suggesting a potential attention shift (e.g., navigating between steps or searching for objects), the system temporarily switches to a burst sampling mode to capture visual information more densely around these moments. In practice, the normal and burst sampling modes trigger a frame pair every 1 s and 0.5 s, respectively, with each pair consisting of two consecutive frames with a 0.1 s interval for optical flow computation (§4.2.2). This design converts always-on low-cost sensing into a motion-aware vision capture strategy that densely samples around critical moments that may require proactive assistance while reducing sampling overhead during stable periods.

4.2.2 Motion-Based Key Moment Selection. Unlike anomaly detection on sensor signals that filters idle moments for improving activity recognition [11], Pro²Assist uses visual motion cues to identify frames that need VLM reasoning for efficient intent inference. Even with adaptive sampling, not all captured frames are equally informative for proactive reasoning. We therefore leverage optical flow, which provides dense motion cues, to further identify frames with meaningful contextual changes. A direct approach would compute optical flow uniformly over the full frame. However, our observations in §3 show that hand manipulations are particularly informative

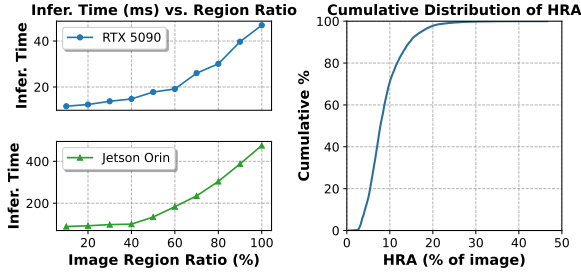


Fig. 9. Optical flow computation time and area ratio distribution of hand-related regions (HRA).

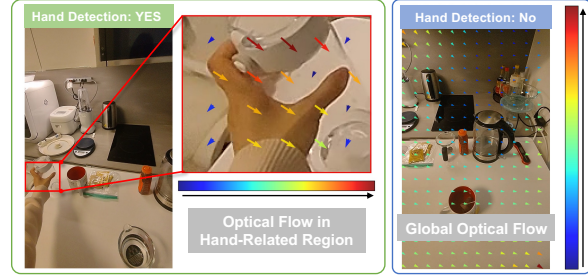


Fig. 10. Optical flow estimation adapts to hand presence. Arrows and colors represent motion direction and magnitude.

of user attention and execution status, which can further provide essential hand motion cues for downstream VLM reasoning (see § 4.3.2 for details). Meanwhile, as shown in Figure 9, optical flow cost scales with region size, and hand regions typically occupy less than 30% of the frame, indicating that restricting optical flow computation to hand-related regions can effectively reduce processing overhead. Thus, as shown in Figure 10, Pro²Assist first detects hands in the field of view, then computes optical flow over hand regions if hands are detected, or over the full frame otherwise. Frames with optical flow magnitude below a motion-based filtering threshold τ_f are filtered out to avoid redundant intent inference, while the remaining frames serve as key moments suggesting a higher likelihood of proactive assistance needs. By focusing on these motion-rich frames, Pro²Assist reduces the processing burden on subsequent VLM reasoning while retaining moments where proactive assistance is needed.

4.3 Step-Oriented Procedural Context Extraction

As discussed in §3 (Observation 3), effective procedural action understanding relies on both temporal context and task-specific expert knowledge. Thus, Pro²Assist proposes a step-oriented procedural context extraction approach for effective modeling of procedural knowledge and temporal dynamics, enabling continuous tracking of fine-grained task progress.

4.3.1 Expert Knowledge Retrieval. Many multi-step physical tasks in daily life, from cooking to assembly, rely on task-specific expert knowledge that is typically provided through instruction guidelines [8]. The right side of Figure 12 illustrates a step graph derived from a procedural guideline that explicitly specifies both sequential and parallel dependencies between procedural steps and step-level execution details. These dependencies and execution details together form the expert knowledge that Pro²Assist integrates to better infer the user's current task state and provide informative, contextually appropriate guidance. Specifically, Pro²Assist takes an initial task instruction $\mathcal{I}$ as input to initiate the service. This instruction could be explicitly provided by the user through speech (e.g., "I'm going to brew a cup of tea. Please provide step-by-step guidance."), as in our work, or implicitly inferred by a one-shot proactive system [74] from environmental context. Next, Pro²Assist retrieves a guideline g_{task} from WikiHow [38], a large-scale public repository of procedural instruction articles in free-form natural language, i.e., $g_{task} = \text{Retriever}(\mathcal{I}, \mathcal{D}_{task})$, where Retriever is the retrieval model. However, directly feeding g_{task} into the system introduces an additional burden during real-time inference, as free-form guidelines leave inter-step dependencies implicit and would require the system to extract structured procedural knowledge from free-form text at every reasoning turn. Pro²Assist therefore offloads this extraction to an advanced LLM (e.g., Claude or GPT series [1, 55]), whose strong language capabilities have been widely validated, as a one-time step at task initiation. The LLM extracts task-specific expert knowledge from the retrieved guideline and transforms it into a structured representation $\mathcal{G} = \text{LLM}(p, \text{examples}, g_{task})$, where p denotes the prompt (details in Appendix A.2) and *examples* are few-shot guideline exemplars demonstrating the desired structured representation, constructed

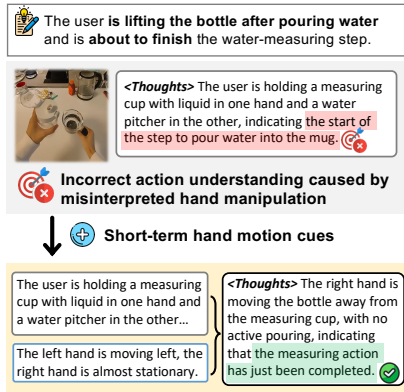


Fig. 11. Impact of incorporating short-term hand motion cues and their impact.

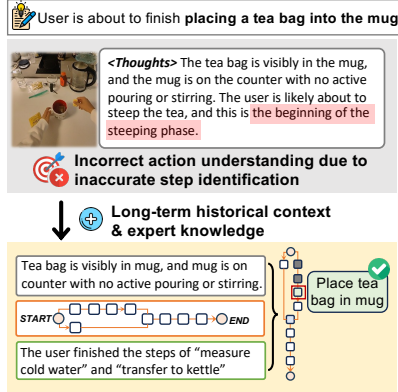
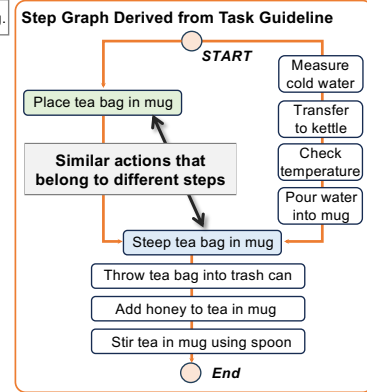


Fig. 12. An illustration of incorporating long-term historical task progress and task-specific expert knowledge.



based on the EgoPER dataset [40]. Pro²Assist then incorporates the structured guideline $\mathcal{G}$ into the VLM prompt to support step-aware reasoning (details are provided in §4.4).

4.3.2 Multi-Scale Temporal Context Extraction. As shown in Figures 11 and 12, relying on single-moment observations makes it difficult to distinguish between different execution statuses within the same step and between similar actions of different steps, indicating that effective procedural action understanding and continuous tracking require temporal context. Such temporal context includes short-term hand manipulations that capture dynamics within a step, and long-term task progress that tracks dynamics across the task workflow. Rather than directly passing dense visual streams to the VLM, which introduces rapidly growing computational overhead (Figure 7), Pro²Assist extracts each scale through compact textual cues for efficient temporal modeling.

Short-term Hand Motion Cues. The optical flow estimated during motion-aware perception (§4.2) provides a natural source of short-term motion cues. A straightforward approach would be to feed optical flow visualizations directly into VLMs. However, existing VLMs typically struggle to interpret optical flow representations directly, as they are primarily trained on RGB data [44, 53]. Moreover, dense per-pixel motion information are redundant for procedural action understanding, which is mainly driven by hand manipulations. Therefore, Pro²Assist extracts motion cues from hand-related regions and represents them in a compact textual form. As shown on the left side of Figure 10, optical flow is estimated within hand-related regions between consecutive frames. Per-pixel flow vectors are then aggregated and decomposed into motion magnitude and direction angle, with the direction angle mapped into one of eight cardinal directions defined on the 2D image plane (up, down, left, right, and four diagonals). The result is converted into natural language descriptions, such as “The left hand is moving down-right, the right hand remains almost stationary.” Pro²Assist incorporates these hand motion cues into the VLM prompt as descriptions of image-plane hand movements (details are provided in §4.4), enabling the VLM to better interpret hand manipulations and improve procedural action understanding, as illustrated at the bottom of Figure 11.

Long-term Historical Task Progress. Incorporating past reasoning traces into the VLM prompt would provide long-term temporal context, but these traces accumulate as the task proceeds, leading to overly long prompts and degraded reasoning performance. Pro²Assist therefore maintains a compact structured text record of completed steps from past verified predictions as the user performs the task, e.g., “[Measure 12 ounces of water, Transfer water to kettle]”. This sequence of completed steps, combined with the retrieved guideline $\mathcal{G}$, effectively indicates task progress and localize the current step. The record is initialized as empty at the start of each task session and progressively updated as verified step transitions occur throughout the task (see §4.4.2 for the historical

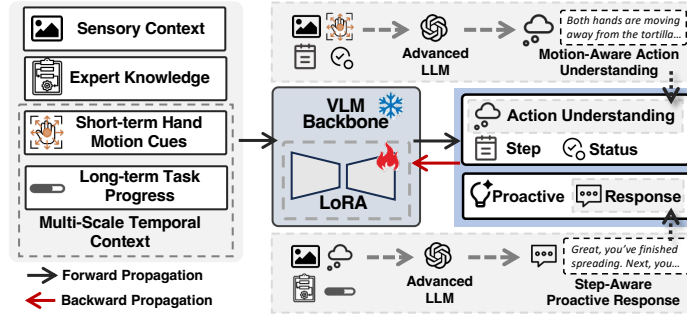


Fig. 13. Training pipeline of the step-aware proactive reasoner in Pro²Assist.

context update mechanism). As shown on the left side of Figure 12, combining this progress record with the retrieved guideline enables the VLM to better identify the user’s current step within the task workflow, thereby distinguishing visually similar actions across different steps and anticipating possible next steps.

4.4 Step-Aware Proactive Reasoner

As discussed in §3 (Observation 1), assistance for procedural tasks requires the system to reason over the user’s evolving state in the task rather than perform single-moment holistic scene understanding. Pro²Assist therefore introduces a step-aware proactive reasoner that jointly reasons over visual input with the extracted procedural context, enabling the system to capture evolving user needs and deliver continuous, step-aware assistance throughout tasks. A consistency checking mechanism further mitigates the impact of single-moment mispredictions and suppresses redundant responses.

4.4.1 VLM Reasoner Training and Inference. Unlike prior work that forecasts next-step assistance and waits for step completion triggers through two parallel processes [41], Pro²Assist uses a single VLM reasoner to continuously reason over the user’s evolving state, deciding whether and what assistance to provide, grounded in the user’s actual state at each inference moment. Achieving this requires the reasoner to learn not only what the user’s state is, but also whether assistance is needed and how to generate responses aligned with that state. Pro²Assist therefore trains the reasoner using a combination of ground-truth supervision with offline distillation from an advanced LLM that provides explicit reasoning chains, equipping it to jointly perform procedural action understanding and step-aware proactive reasoning.

Figure 13 illustrates the overall training pipeline. During training, each instance consists of multimodal inputs and hierarchical supervision targets. The inputs consist of the egocentric image, structured guideline, and multi-scale temporal context. The supervision targets include ground-truth step, execution status, and proactive trigger labels, along with LLM-generated motion-aware action understanding and step-aware proactive responses as distillation targets. All LLM-generated annotations are validated by human annotators for quality and consistency (details in Appendix A.1). The distillation targets equip the reasoner with two key capabilities. For motion-aware action understanding, the advanced LLM generates explicit reasoning chains that map visual and motion cues to the corresponding step and execution status, teaching the reasoner to interpret short-term hand motion cues. For step-aware proactive response generation, the LLM generates high-quality step-aware responses, demonstrating how to align guidance with the user’s current state and procedural knowledge. Through supervised training with distillation, the VLM reasoner learns to reproduce both behaviors while predicting ground-truth labels.

During inference, guided by the system prompt shown in Figure 14, the reasoner identifies the step with execution status and determines whether assistance is needed. If no assistance is required, it remains silent to

Template of System Prompt	
Task Instructions: You are a proactive assistant for procedural tasks, tracking user progress through real-time sensory data from wearable devices and procedural context. Based on these contexts, you need to understand user's actions and identify the current procedural step and status, then determine whether to initiate a proactive service for the user or not. Please provide timely, relevant guidance when appropriate. Now you will receive the sensory and procedural contexts for the target moment in the procedural task. Guideline: Structured guideline with task-specific expert knowledge: $\langle \text{GUIDELINE} \rangle$ Historical Context: Previously completed steps in order: $\langle \text{STEP-1} \rangle, \langle \text{STEP-2} \rangle, \dots$ Sensory Context: Vision information shows as the image. $\langle \text{IMAGE} \rangle$ Hand Motion Cues: The optical flow between the current frame and the previous frame indicates that in the image plane, $\langle \text{HAND_MOTION_CUES} \rangle$. You may combine the image-plane hand motion cue with visual context, object relationships, and task semantics to infer how the hand is moving in the actual scene when useful.	Input: Predicted step $\hat{D}_s$, execution status $\hat{S}_s$, proactive trigger $\hat{P}_l$, and proactive response $\hat{P}_r$ at moment t State: Historical task progress $\hat{H}_p$ (an ordered list, initially empty); active step D_s^{act} (initially the first predicted step); sliding window $\mathcal{B}$ (max size W); previously predicted step D_s^{prev} and status S_s^{prev} (both initially empty) // Mechanism 1: Historical Context Update 1: Append $\hat{D}_s$ to $\mathcal{B}$; remove oldest if $\mathcal{B} > W$ 2: $D_s^{\text{dom}} \leftarrow$ most frequent step in $\mathcal{B}$ 3: if $D_s^{\text{dom}} \neq D_s^{\text{act}}$ and $\text{count}(D_s^{\text{dom}}, \mathcal{B}) \geq \tau_c \cdot \mathcal{B}$ then 4: Append D_s^{act} to $\hat{H}_p$ ▷ mark completed 5: $D_s^{\text{act}} \leftarrow D_s^{\text{dom}}$ // Mechanism 2: Response Delivery Control 6: if $\hat{P}_l = \text{true}$ then 7: if $\hat{D}_s = D_s^{\text{prev}}$ and $\hat{S}_s = S_s^{\text{prev}}$ then 8: $\hat{P}_r \leftarrow \text{None}$ ▷ suppress redundant 9: else 10: Deliver $\hat{P}_r$ ▷ display response 11: else 12: $\hat{P}_r \leftarrow \text{None}$ ▷ no proactive need 13: $D_s^{\text{prev}} \leftarrow \hat{D}_s$; $S_s^{\text{prev}} \leftarrow \hat{S}_s$ Output: Updated $\hat{H}_p$; $\hat{P}_r$ (or None if suppressed)

Fig. 14. System prompt for the step-aware proactive reasoner in Pro²Assist.

Algorithm 1. Step-aware consistency checking, including historical context update and response delivery control.

avoid unnecessary interruption. Otherwise, it generates a step-aware proactive response aligned with the user's state to guide them in completing the current step or proceeding to the next.

4.4.2 Step-Aware Consistency Checking. While Pro²Assist leverages temporal context for continuous reasoning rather than treating each prediction independently, single-moment mispredictions can occur and accumulate over time. Moreover, repeatedly delivering similar responses for the same user state can be disruptive and distract the user from the task at hand. To ensure reliable and non-intrusive assistance, as shown in Algorithm 1, Pro²Assist proposes a step-aware consistency checking mechanism that handles the reasoner's predictions, benefiting both long-term temporal context update and response delivery.

Historical Context Update Mechanism. Maintaining long-term historical task progress based on past observations and predictions is critical for continuous reasoning. However, directly incorporating every single-moment prediction into the historical record would propagate mispredictions and degrade subsequent reasoning. Unlike passively decaying memory items based on temporal scores [51], Pro²Assist therefore uses a sliding window as a temporal consistency check over historical step predictions, updating the record only when step transitions are consistently observed across multiple moments. Specifically, when the predicted step $\hat{D}_s$ differs from the active step D_s^{act} , Pro²Assist treats this as a candidate transition rather than immediately updating the historical record $\hat{H}_p$. It then evaluates predictions within the sliding window $\mathcal{B}$, adding D_s^{act} to the record as completed only when the new step appears in the majority of $\mathcal{B}$. This update mechanism mitigates the impact of single-moment mispredictions and ensures that only temporally consistent step transitions are reflected in the historical task progress, providing more reliable long-term context for subsequent reasoning.

Response Delivery Control. Users often remain in the same step and execution status for some time. Repeating similar proactive assistance under such stable states provides little additional benefit and could distract the

user from the ongoing task. To avoid repetitive interruptions, Pro²Assist suppresses response delivery when the predicted step and execution status ($\hat{D}_s, \hat{S}_s$) match those from the previous moment ($D_s^{\text{prev}}, S_s^{\text{prev}}$), indicating that the user is during steady execution and does not need further intervention. This mechanism mitigates redundant responses and ensures that proactive assistance is delivered when meaningful state changes occur.

5 EVALUATION

In this section, we first introduce the system implementation and the datasets used for evaluation, including a dataset curated from public sources and a real-world dataset collected using our testbed. We then introduce the evaluation metrics and baseline methods. Finally, we present the evaluation results of Pro²Assist together with findings from a user study.

5.1 System Implementation

5.1.1 Testbed Setup. We implement Pro²Assist on a real-world hardware testbed consisting of RayNeo X3 Pro [3] smart glasses and a back-end server. The RayNeo X3 Pro integrates a Sony IMX681 camera, a built-in IMU, a four-speaker audio system, and a dual micro-LED projector system. Its battery supports up to 5 hours of continuous operation, which is sufficient to run Pro²Assist throughout typical procedural tasks. A Kotlin-based Android client runs on the glasses to collect multimodal sensor data (vision, IMU, and audio), transmit it to the server over WiFi 6, and display assistance on the glasses. Pro²Assist is evaluated on three back-end platforms, including NVIDIA Jetson Orin and two servers with NVIDIA RTX 5090 and RTX PRO 6000 GPUs, respectively. Lightweight models are implemented using PyTorch [50], and we use Ollama [2] for VLM inference.

5.1.2 Configuration. In the experiments, we fine-tune VLMs using low rank adaptation (LoRA) [33] with a rank of 8, training for 10 epochs at a learning rate of 5×10^{-4} on an NVIDIA RTX PRO 6000 GPU. We fine-tune YOLO11n [37] on the EgoPER dataset [40] for hand detection, and employ the RAFT model [57] for optical flow estimation. The sampling threshold τ_s and the filtering threshold τ_f are set to 0.3 and 10, respectively. For expert knowledge retrieval, we use Azure Speech Recognition [48] for speech-to-text conversion and all-mpnet-base-v2 [7] for guideline retrieval, with embeddings of WikiHow articles precomputed and stored locally. The advanced LLMs used in Pro²Assist are GPT-5 [55] and Claude-Sonnet-4.5 [1]. For evaluation, we deploy Pro²Assist with seven VLMs of varying scales, including Qwen3-VL series (2B/4B/8B/30B) [13] and InternVL3 series (2B/8B/14B) [22]. By default, we use Qwen3-VL-4B, and evaluate all baselines at the same scale unless otherwise specified.

5.2 Experimental Setup

5.2.1 Dataset. Although existing procedural datasets contain egocentric videos [15, 36, 40, 43], there is no dataset containing proactive assistance annotations that can be directly used for evaluation. Therefore, we first curate a dataset by augmenting public procedural datasets with annotations for proactive procedural assistance. In addition, we collect a real-world dataset of multimodal data (egocentric video, IMU, and audio) using our testbed. **Proactive Procedural Dataset.** We construct a dataset by augmenting public egocentric procedural data with fine-grained annotations of procedural action understanding and step-aware proactive assistance. These annotations are generated jointly by human annotators and advanced LLMs.

Egocentric Visual Data Source. The public sources used to curate our dataset are as follows.

- GTEA dataset [30] contains videos of seven daily activities performed by four participants, captured using a GoPro camera mounted on a cap at 30 FPS. In experiments, we use the released rectified data provided at 15 FPS. We utilize three relatively complex tasks with five procedural steps each (making hotdog, cheese sandwich and peanut-butter sandwich) from it, and randomly sample videos from three participants for each task.

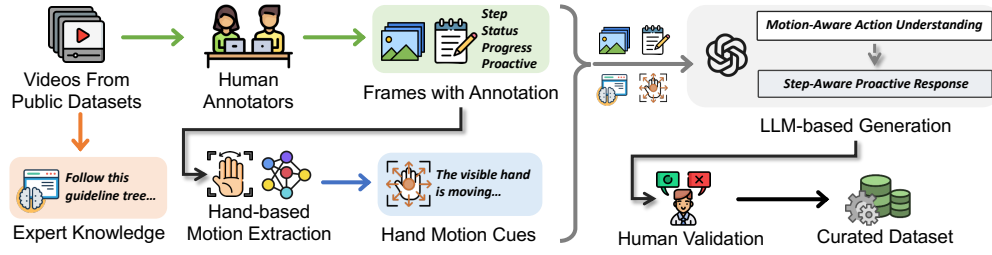


Fig. 15. Our dataset curation pipeline combining human annotation and LLM-assisted generation.

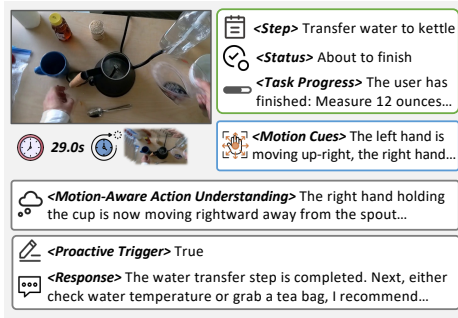


Fig. 16. A sample from the curated dataset, consisting of an egocentric frame and its corresponding annotations.

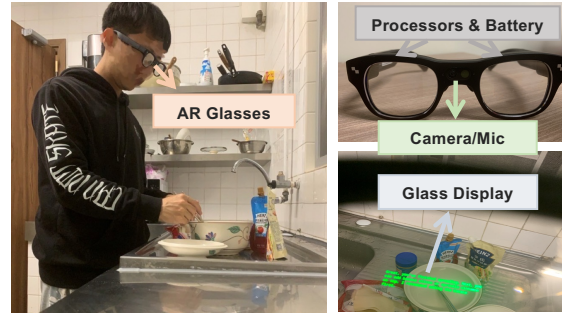


Fig. 17. Illustration of the real-world evaluation testbed. Participants wear AR glasses while performing procedural tasks, and Pro²Assist displays step-aware proactive messages directly on the glasses.

- EgoPER dataset [40] consists of egocentric procedural task videos with step annotations collected from 11 participants wearing Microsoft HoloLens2 at 15 FPS. It includes five tasks (making pinwheels, quesadilla, oatmeal, coffee, and tea), with more than ten steps on average per task. For each task, we randomly select videos from three to four participants with different execution orders.
- EgoProceL dataset [15] contains egocentric videos with key-step annotations for procedural learning, constructed from both public datasets and self-collected videos, with sampling rates between 12 FPS and 60 FPS. We utilize two daily procedural tasks (tent assembly and personal computer assembly) from the dataset, and randomly sample videos from three participants for each task.

After collecting videos with step annotations from these datasets, unless otherwise specified, we sample frames from the raw videos at 10 FPS, which is commonly used in video understanding tasks for procedural tasks [49, 54].

Sample Annotation Procedure. As shown in Figure 16, each sample the curated dataset consists of $\mathcal{S} = \{I_t, t, I_{prev}, D_s, S_s, H_p, M_h, A_u, P_l, P_r\}$, where I_t is the current frame, t is the timestamp, $I_{prev} = \{I_{t-9}, \dots, I_{t-1}\}$ are the previous nine frames, and the remaining terms denote step description, execution status, historical task progress, hand motion cues, motion-aware action understanding, proactive trigger, and step-aware proactive response, respectively. The step description D_s is obtained from the original key-step annotations in public datasets. As shown in Figure 15, the rest are generated through a two-stage procedure that combines human expertise with LLM-assisted generation. Detailed procedures are in Appendix A.1.

In total, our curated dataset² contains 1,089 samples covering ten common daily procedural tasks (i.e., making coffee, oatmeal, tea, pinwheels, quesadilla, cheese sandwich, hotdog, peanut butter sandwich, as well as PC

²The curated dataset is available at <https://github.com/Columbia-ICSL/Pro2Assist>.

assembly and tent assembly). For supervised fine-tuning (SFT) in experiments, we randomly split the dataset into 65% for training and 35% for testing.

Real-world Evaluation. While our curated dataset provides annotated egocentric visual data for proactive procedural assistance, it lacks paired multimodal sensor streams required for end-to-end system evaluation. Therefore, we conduct real-world experiments and a user study on our real-world testbed. Specifically, we recruited 20 participants to perform assigned procedural tasks. Figure 17 illustrates the testbed and data collection setup. Each participant wears RayNeo X3 Pro smart glasses to capture synchronized multimodal data during the task. The study was approved by the authors’ institutional IRB, and all participants provided informed consent. After data collection, participants review their recorded videos, segment them into temporal segments, and annotate each segment with step descriptions (D_s), step status (S_s), and proactive labels (P_l). They also participated in a user study and completed a questionnaire based on their experience. In total, we collected 20 multimodal recordings with an average duration of 325.4 s, covering five different procedural tasks from the curated dataset (i.e., making tea, quesadilla, cheese sandwich, hotdog and peanut butter sandwich).

5.2.2 Evaluation Metrics. We extensively evaluate Pro²Assist’s performance from five perspectives as follows.

Procedural Action Understanding Accuracy. This dimension assesses the system’s ability to correctly infer *which step* the user is performing and *how far along* they are within that step (i.e., execution status). *Step-Acc* (Step Identification Accuracy) measures whether the predicted step matches the ground truth, and *Status-Acc* (Status Identification Accuracy) evaluates whether the fine-grained execution status (i.e., *just start*, *in progress*, *about to finish*, and *step transition*) is correctly identified. For example, given the sample in Figure 16, *Step-Acc* evaluates whether the system correctly identifies the current step as “Transfer water to kettle” rather than other steps in the procedural task (e.g., “Measure 12 ounces of water”), and *Status-Acc* evaluates whether it correctly recognizes the status as *about to finish* rather than other execution statuses.

Proactive Trigger Performance. This aspect evaluates whether proactive assistance is initiated at appropriate moments. Following prior work [74], we use *Acc-P* (Proactive Accuracy) to measure the correctness of proactive trigger predictions, *MD* (Missed Detection) to quantify the rate of failing to trigger assistance when needed, and *FD* (False Detection) to measure the rate of incorrectly triggering assistance when not needed.

Proactive Timing Accuracy. The timing of step-aware proactive assistance is critical, as responses delivered too late may fail to help the user, while responses triggered at incorrect steps provide little value. Moment-level metrics (e.g., *Acc-P*, *MD*, *FD*) only evaluate whether the system correctly triggers proactive assistance, but do not capture *how timely* the response is within the valid proactive period. In procedural tasks, a response triggered early in the proactive period is more useful than one triggered at the last moment. To evaluate timing quality, we introduce the *Step-aware Timeliness Score (STS)*, a period-level metric ranging from 0 to 1, where a higher score indicates earlier and more useful proactive assistance. For a predicted trigger time $\hat{t}_i$ with predicted step $\hat{D}_{s_i}$, given ground-truth step D_{s_i} and the corresponding proactive time period $[s_i, e_i]$, STS is defined as

$$STS_i = \begin{cases} \exp\left(-\frac{\hat{t}_i - s_i}{e_i - s_i}\right) & \text{if } \hat{D}_{s_i} = D_{s_i} \text{ and } s_i \leq \hat{t}_i \leq e_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

This metric assigns higher scores to earlier responses within the valid proactive period for the correct step, with scores decaying exponentially toward the end of the period. A response at the final moment of the period ($STS \approx 0.368$ when $\hat{t}_i = e_i$) is still valued more than a missed or misaligned response ($STS = 0$). Overall, STS is computed by averaging across all proactive moments. By considering temporal position within the valid proactive period, STS goes beyond binary trigger correctness and captures the practical utility of proactive assistance.

Response Quality. To measure the relevance and usefulness of generated proactive responses, we adopt an LLM-as-a-Judge approach, whose effectiveness has been demonstrated in prior work [19, 32]. We incorporate the

ground-truth step, execution status, and task-specific guideline into an evaluation prompt, and use GPT-5 [55] to assess response quality. Prompt details are provided in Appendix A.2.

System Overhead. We measure system overhead from four aspects to evaluate Pro²Assist’s efficiency for real-world deployment. *Inference Ratio* denotes the proportion of moments in which the VLM reasoner is invoked for step-aware proactive reasoning. *Proactive Hit Rate* measures the percentage of VLM inferences that occur within ground-truth proactive periods. These two metrics should be interpreted together, as a lower *Inference Ratio* is desirable only when *Hit Rate* remains high, indicating selective yet effective trigger of VLM inference. *Latency* includes inference latency and communication delay. *Power Consumption* measures the average power usage of the smart glasses during system operation.

5.2.3 Baselines. We evaluate Pro²Assist and compare it with strong baselines that are adapted to the proactive procedural assistance setting, including an existing procedural assistant extended with proactive capabilities (VLM-Procedure), few-shot prompting strategies adapted for proactive procedural reasoning (Vanilla ICL, ICL-EN, and CoT), and existing general proactive systems adapted to procedural tasks (VideoLLM-online and ProAgent).

VLM-Procedure. PrISM-Q&A [10] is an LLM-based reactive procedural assistant originally designed to operate using audio, IMU, and task knowledge. We employ its system prompt and extend it to support VLM-based reasoning and visual inputs. We further adapt the prompt to support proactive reasoning, enabling the model to perform proactive reasoning and generate proactive assistance.

Vanilla ICL. This baseline uses in-context learning (ICL) [25] with few-shot demonstrations that contain only raw sensory context, relying on the VLM’s intrinsic knowledge to perform action understanding and proactive reasoning for multi-step procedural tasks. It serves as a minimal baseline for the prompting-based approaches. Together with ICL-EN and CoT, these baselines evaluate whether widely adopted prompting techniques can sufficiently address proactive procedural assistance.

ICL-EN. Built upon the Vanilla ICL baseline, this approach explicitly incorporates task-specific expert knowledge into the system prompt, providing additional guidance for understanding procedural actions. This evaluates to what extent expert knowledge injection via prompting can further improve performance.

CoT. This approach employs a concise Chain-of-Thought [64] strategy with few-shot examples containing explicit thought traces that demonstrate how to map visual cues to the current procedural step, its execution status, and step-aware proactive assistance. This evaluates to what extent explicit structured reasoning via prompting can further improve performance.

VideoLLM-online. VideoLLM-online [20] is an online VLM designed for streaming video, which introduces a Streaming EOS (End-of-Sequence) prediction objective at the model level to enable proactive response updates. We use the released VideoLLM-online-8B-v1+ model as a baseline in our real-world evaluation, and primarily compare it with Pro²Assist on proactive prediction performance. Together with ProAgent, these baselines evaluate whether existing proactive methods can generalize to long-horizon procedural tasks.

ProAgent (Vanilla&FT). ProAgent [73] is a proactive assistance system designed for general daily scenarios based on holistic scene understanding. For a comprehensive comparison, we evaluate ProAgent under two configurations. ProAgent (Vanilla) uses the original model trained on the CAB-Lite dataset [74], following its original settings. ProAgent (FT) is further fine-tuned on our curated dataset to better adapt it to procedural tasks. Since ProAgent is not explicitly designed for procedural action understanding, we primarily compare it with Pro²Assist on proactive prediction performance.

For baselines without task-specific expert knowledge, we provide the complete set of possible steps in the curated dataset to ensure fair step identification. For few-shot demonstrations, we randomly include five examples from the dataset in the prompt. In the real-world evaluation, we implement VLM-Procedure with periodic sampling at 0.5 s intervals to enable continuous perception and proactive reasoning. For the ICL, ICL-EN, and CoT baselines, visual data are sampled at 10 FPS and processed by Reducto [42] to remove redundant frames

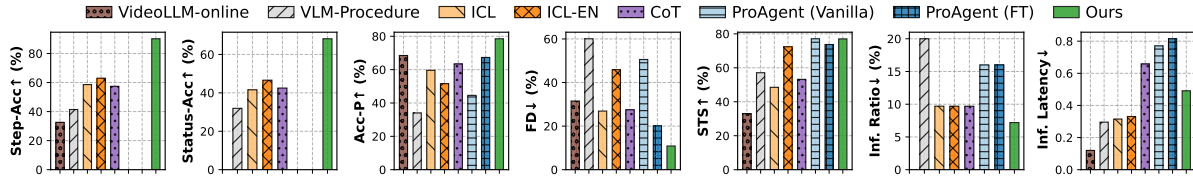


Fig. 18. End-to-end performance comparison in real-world evaluation. Missing bars indicate that the corresponding metric is not applicable to that method due to its design.

before VLM inference, improving efficiency in real-world settings. For ProAgent, we adapt its on-demand tiered perception to procedural tasks by setting the low-rate and high-rate sampling intervals to 1 s and 0.5 s, respectively.

5.3 Overall Performance

5.3.1 Quantitative Results. In this section, we evaluate the overall performance of Pro²Assist on both the real-world testbed and the curated dataset.

On real-world testbed. As shown in Figure 18, we compare Pro²Assist with multiple baselines on the real-world evaluation. While VideoLLM-online achieves low inference latency and 68.4% *Acc-P*, it performs poorly on procedural action understanding, with only 32.6% *Step-Acc*, which further degrades proactive timing accuracy to 33.0% *STS*. This is because VideoLLM-online is specifically designed to efficiently process dense video streams, but does not model user intent and procedural knowledge that are essential for proactive assistance in procedural tasks. VLM-Procedure achieves better procedural action understanding with expert knowledge, but performs poorly in proactive reasoning when relying on the model’s intrinsic knowledge to provide proactive assistance, indicating that prompt-level extension of a reactive procedural assistant cannot acquire effective proactive capability. Prompting techniques produce partial improvements with limited overall gains. Specifically, among the prompting baselines, ICL-EN and CoT exhibit distinct improvement patterns over Vanilla ICL. Specifically, ICL-EN primarily improves procedural action understanding (4.4% *Step-Acc*, 5.1% *Status-Acc*) but decreases *Acc-P*, while CoT primarily improves proactive prediction (3.8% *Acc-P*) without improving procedural action understanding, demonstrating that incorporating expert knowledge and reasoning traces via prompting benefits different parts of the proactive procedural task. However, the improvements among prompting baselines remain confined to 3–5%, and the overall performance of all three remains limited, indicating that widely adopted prompting techniques alone are insufficient for proactive procedural assistance. ProAgent (Vanilla), which is primarily designed for general daily scenarios, exhibits limited performance on long-horizon procedural tasks, as it lacks explicit designs for procedural knowledge modeling and capturing temporal context in procedural tasks. Even after fine-tuning on the curated dataset, ProAgent (FT) still underperforms Pro²Assist, as it lacks explicit designs for procedural knowledge and temporal context, both of which are critical for procedural tasks. Since ProAgent cannot reliably identify the current procedural step, we relax the step-matching condition in the *STS* computation for ProAgent. However, ProAgent still achieves a lower *STS* than Pro²Assist. Overall, compared to the best-performing baselines, Pro²Assist still obtains improvements of 25.2% in *Step-Acc*, 21.6% in *Status-Acc*, and 15.1% in *Acc-P*. Moreover, it achieves up to 2.29× the *STS* of the baselines. Pro²Assist maintains an inference latency within 0.5 s, slightly higher than baselines such as ICL but with a substantially lower VLM inference ratio, demonstrating an effective trade-off between efficiency and performance. Together, these results show that no existing approach captures all the capabilities required for proactive procedural assistance, whereas Pro²Assist addresses them jointly through its integrated design, validating its effectiveness for real-world procedural task assistance.

On the curated dataset. As shown in Figure 19, we further evaluate Pro²Assist and baseline methods on the curated dataset, focusing on the overall capability of the VLM reasoner for procedural action understanding and

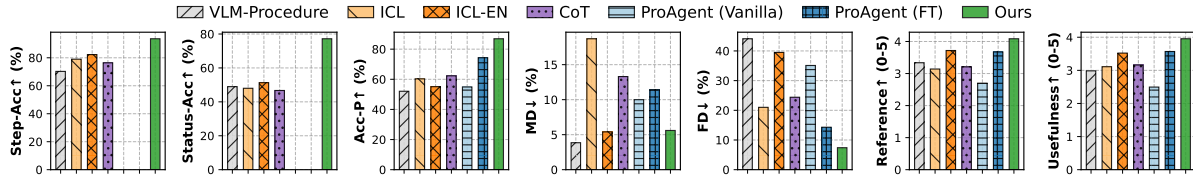


Fig. 19. Overall performance comparison on the curated dataset. Missing bars indicate that the corresponding metric is not applicable to that method due to its design.

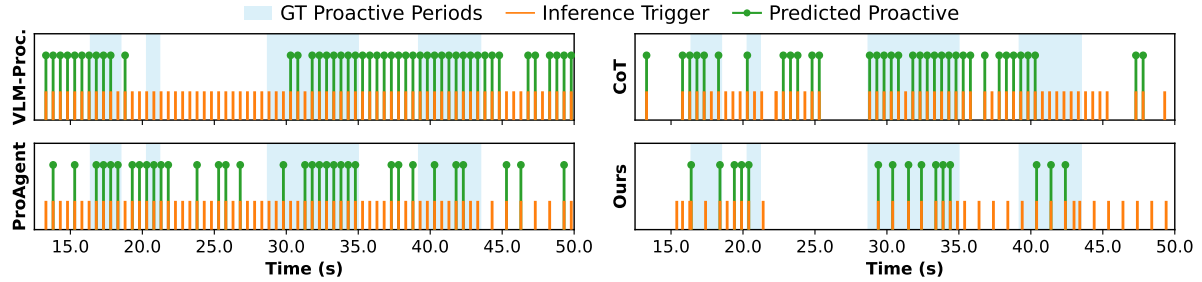


Fig. 20. Performance comparison of inference triggering and proactive prediction. Solid orange lines indicate VLM inference timestamps, blue shaded regions denote ground-truth proactive intervals, and green lines with circular markers represent predicted proactive triggers. “VLM-Proc.” represents the VLM-Procedure baseline.

step-aware proactive reasoning. Overall, Pro²Assist significantly outperforms all baselines across all evaluation metrics. In particular, Pro²Assist achieves 93.6% *Step-Acc* and 77.2% *Status-Acc*, indicating effective procedural action understanding. Besides, It achieves 86.9% *Acc-P* with *MD* and *FD* both below 8%, demonstrating its ability to accurately identify moments requiring proactive assistance while avoiding unnecessary or premature interventions. Moreover, Pro²Assist achieves the highest scores in both *Reference* and *Usefulness*, indicating its assistance is not only timely but also contextually appropriate and useful.

5.3.2 Qualitative Results. As shown in Figure 20, compared with baselines, Pro²Assist not only triggers VLM inference at appropriate moments with high proactive demand, but also more accurately identifies proactive moments to deliver timely assistance. In contrast, baselines either trigger inference excessively, resulting in redundant predictions, or fail to reliably identify proactive moments, leading to high missed detection and false detection rates. Figure 21 further presents representative examples of Pro²Assist’s inference results on a task recording, demonstrating that Pro²Assist explicitly reasons over sensory and procedural contexts to achieve reliable action understanding. When no assistance is needed, Pro²Assist remains silent to avoid interrupting the user. Otherwise, it generates step-aware assistance based on the user’s current state and expert knowledge to effectively help the user perform tasks.

5.4 Effectiveness of System Module

5.4.1 Impact of Motion-based Perception. We evaluate Pro²Assist’s motion-based perception strategy from two perspectives, including system overhead and overall prediction performance. As shown in Figure 22, we first evaluate the impact of this strategy on system overhead by removing head motion-aware sampling (“w/o Sampling”), motion-based key moment selection (“w/o Selection”), and both components (“w/o Both”). The “w/o Selection” and “w/o Both” variants have high inference ratios with low hit rates, indicating frequent but ineffective VLM inference. The “w/o Sampling” variant yields a lower *Inference Ratio* but also a lower *Hit Rate*,

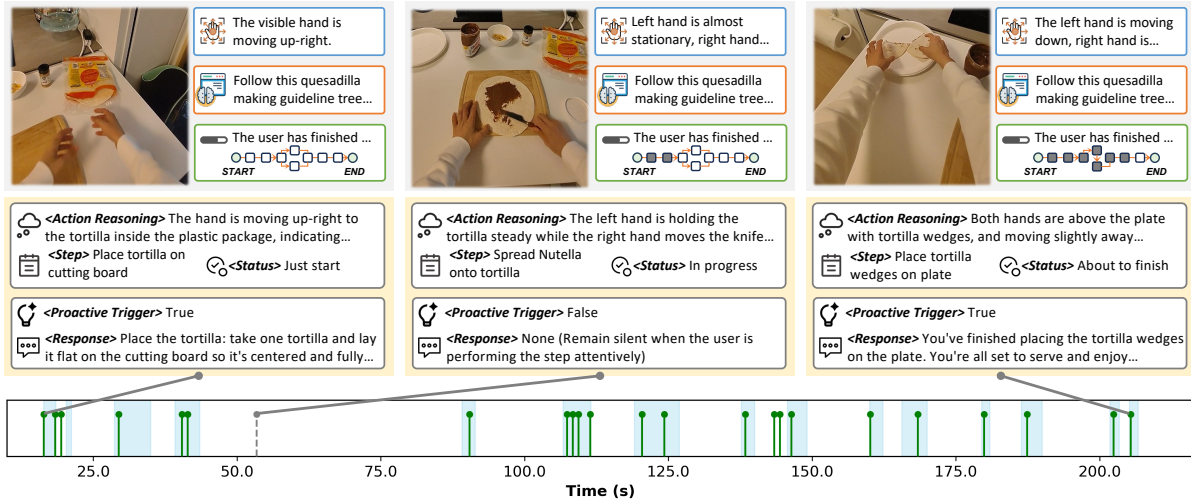
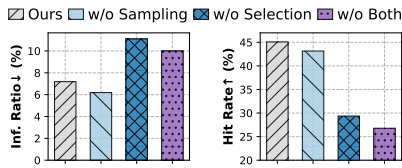
Fig. 21. Examples of Pro²Assist's inference results in real-world evaluation.

Fig. 22. Impact of motion-based perception on inference ratio and proactive hit rate.

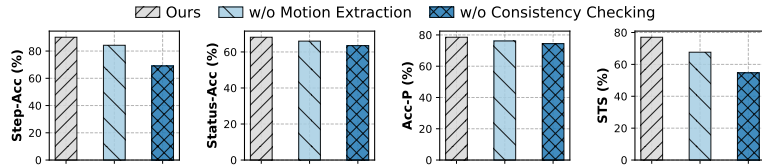


Fig. 23. Impact of motion extraction in motion-based perception and the step-aware consistency checking mechanism.

as uniform-interval sampling fails to capture key moments indicated by head motion that require proactive assistance, which is further supported by the lower values of both metrics for “w/o Both” compared to “w/o Selection”. In contrast, Pro²Assist achieves the best trade-off with the highest *Hit Rate* and a slightly higher *Inference Ratio*, indicating its effectiveness. Furthermore, we evaluate the impact of motion extraction in motion-based key moment selection on overall prediction performance. As shown in Figure 23, compared with the variant without motion extraction (“w/o Motion Extraction”), Pro²Assist achieves improvements across metrics, indicating that incorporating motion extraction allows Pro²Assist to leverage fine-grained hand motion cues for more accurate procedural action understanding, which in turn enhances step-aware proactive prediction.

5.4.2 Impact of Motion-Aware Action Understanding. As shown in Figure 24, removing the motion-aware action understanding objective during SFT (“w/o Action Understanding”) degrades Pro²Assist’s performance on the curated dataset by 2.1% in *Step-Acc*, 6.8% in *Status-Acc*, and 6.2% in *Acc-P*, demonstrating its effectiveness for procedural proactive assistance.

5.4.3 Impact of Temporal Context and Expert Knowledge. We examine variants that remove temporal context, expert knowledge, and both components (denoted as “w/o Temporal Context”, “w/o Expert Knowledge”, and “w/o EK+TC”, respectively). As shown in Figure 24, all the variants lead to significant performance degradation. For example, Pro²Assist achieves improvements of 19.8% in *Step-Acc*, 3.4% in *Status-Acc*, and 5.4% in *Acc-P* over the “w/o EK+TC” variant. Overall, the results validate the effectiveness of incorporating them into reasoning.

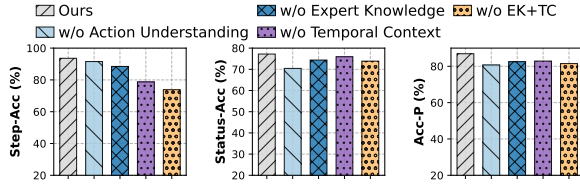


Fig. 24. Ablation study of Pro²Assist's VLM reasoner. "EK" and "TC" denote expert knowledge and temporal context.

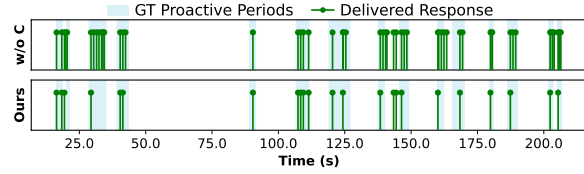


Fig. 25. Impact of step-aware consistency checking on avoiding repeatedly delivering similar assistance.

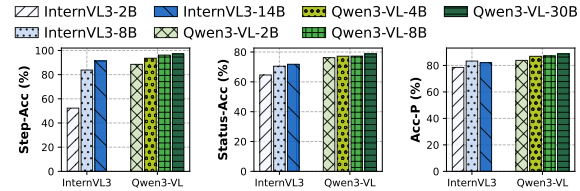


Fig. 26. Comparison of different base VLMs used in Pro²Assist on the curated dataset.

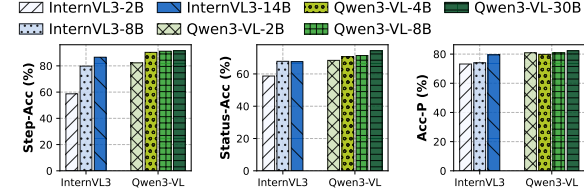


Fig. 27. Performance of Pro²Assist on the curated dataset in out-of-domain settings.

5.4.4 Impact of Step-Aware Consistency Checking. We evaluate the mechanism by removing it ("w/o Consistency Checking") in the real-world evaluation. First, we evaluate its effectiveness in reducing unnecessary user interruptions. As shown in Figure 25, with the mechanism, Pro²Assist delivers new assistance only when the user's state changes, avoiding repetitive guidance and reducing perceived intrusiveness by over 50%. Second, we evaluate its effectiveness in preventing single-moment mispredictions from degrading overall performance. As shown in Figure 23, removing it results in a drop of over 20% in both *Step-Acc* and *STS*, indicating that single-moment mispredictions accumulate and degrade subsequent reasoning. These results demonstrate that the mechanism is crucial for both user experience and robust reasoning.

5.4.5 Impact of Hyper-parameters. We further evaluate the effects of parameter settings.

Parameter Sensitivity in Motion-Based Perception We analyze the impact of the sampling and filtering thresholds on *Inference Ratio* and *Proactive Hit Rate*, which jointly evaluate the tradeoff between computational efficiency and sampling effectiveness. As shown in Figure 28, we sweep one threshold while fixing the other. For motion-aware sampling, a low threshold cannot help with avoiding unnecessary VLM inference, while an overly large threshold makes the system miss attention shifts indicated by head motion, reducing hit rate. Similarly, a low filtering threshold keeps many frames with minimal motion, while an excessively high threshold may filter out frames requiring assistance, negatively affecting proactive performance. Within a reasonable range of threshold values, both metrics remain stable, reflecting the robustness of the perception mechanism, which effectively combines low-cost, always-on head motion signals with fine-grained hand motion cues. Overall, across a wide range of settings, Pro²Assist consistently outperforms VLM-Procedure with periodic sampling (20% inference ratio, 27% hit rate), indicating that motion-based perception provides a better efficiency–effectiveness tradeoff.

Impact of Base VLM Models. As shown in Figure 26, we further evaluate Pro²Assist with different VLM models as the base model. The results demonstrate that Pro²Assist works effectively across different base VLMs, and scaling up the base model consistently improves its performance.

Impact of Temporal Window Length in Consistency Checking. We vary the window length in the real-world evaluation. As shown in Figure 29, across all settings, Pro²Assist consistently and significantly outperforms the variant without this checking mechanism (denoted as "w/o Checking"), indicating the effectiveness and robustness of Pro²Assist with respect to the choice of consistency history length.

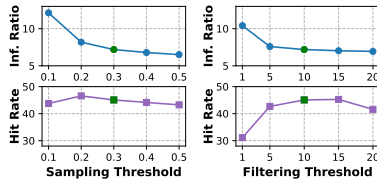


Fig. 28. Impact of threshold settings in motion-based perception.

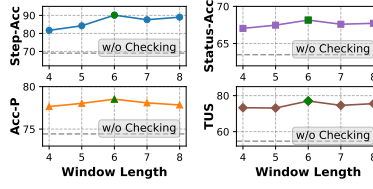


Fig. 29. Impact of window length in step-aware consistency checking.

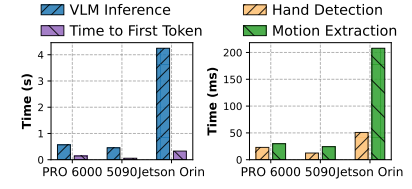


Fig. 30. Pro²Assist's system latency across different devices.

5.5 Out-of-Domain Evaluation

We evaluate Pro²Assist's ability to generalize to unseen tasks by randomly splitting the curated dataset at the procedural task level. Samples from six tasks are used for training, while the remaining four tasks are reserved for evaluation. As shown in Figure 27, Pro²Assist remains effective across different base VLMs on unseen tasks. We further assess Pro²Assist on a subset of the real-world dataset that includes three procedural tasks unseen during training (i.e., making tea, quesadilla, and cheese sandwich), where it achieves on average 82.2% *Step-Acc*, 71.0% *Status-Acc*, 75.3% *Acc-P*, and 67.2% *STS*, indicating its generalization in real-world settings.

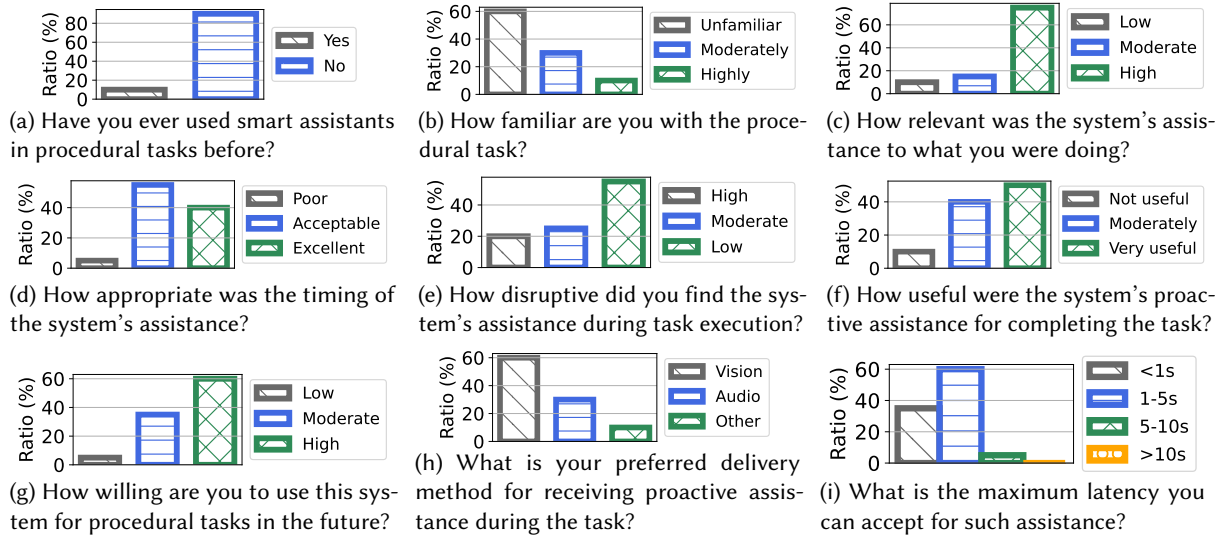
5.6 System Overhead

We evaluate the system overhead of Pro²Assist across multiple back-end platforms. The total inference time consists of VLM inference, hand detection, and motion extraction. As shown in Figure 30, Pro²Assist achieves a total inference time of 0.49 s on an NVIDIA RTX 5090 GPU and 4.51 s on an NVIDIA Jetson Orin. Notably, the time to first token (TTFT) consistently remains below 330 ms, enabling Pro²Assist to begin delivering assistance promptly via streaming output for real-time responsiveness, even on resource-constrained edge devices. Moreover, expert knowledge retrieval takes under 0.22 s across all devices, and the average communication latency in two real-world evaluation environments is 327.3 ms. In addition, we measure the average power consumption of the smart glasses while running Pro²Assist, which is 2.2 W, indicating it is practical for real-world deployment.

5.7 User Study

We conduct a user study with 20 participants (10 male and 10 female, P1-P20) with an average age of 27, whose education levels ranged from undergraduate to Ph.D., to evaluate whether Pro²Assist meets expectations as a proactive procedural assistant. Participants provided feedback for Pro²Assist's assistance through a questionnaire containing three parts with nine questions in total. Following prior studies [28, 47, 70], we applied categorical ratings with distributional analysis for characterizing users' perception of the system, with questions and question-specific response options phrased in plain language, so that participants can interpret each option directly regardless of technical background. The questionnaire contains three parts that capture participants' background, subjective system evaluation, and preferences for proactive procedural assistants, with the specific questions chosen based on Pro²Assist's design as an AR glasses-based assistant and key dimensions adopted by prior proactive and procedural assistance studies [35, 51, 70]. Details are as follows.

- **S1. Background Information.** This part collects participants' prior experience with the task and frequency of performing it, as well as their previous use of smart assistants in procedural tasks.
- **S2. System Evaluation.** *Contextual Relevance* assesses whether the delivered messages match the participant's current step and execution status. *Timeliness* assesses whether messages are delivered at appropriate moments. *Usefulness* assesses whether messages help participants complete the task. *Intrusiveness* assesses whether the system is disruptive due to excessive messages. *Willingness* assesses participants' willingness to use the system in the future.

Fig. 31. Results of the overall user perception for Pro²Assist.

- **S3. System Preferences.** This part examines user preferences for proactive assistant design, including acceptable response latency and preferred delivery method for proactive assistance.

Overall User Perception of Pro²Assist. Figure 31 demonstrates participants' feedback. Among the participants, 60% had no prior experience with the assigned task, and only two had previously used smart assistants for procedural activities. Overall, 90% participants found the system useful, with particularly strong agreement among those without prior task experience. This indicates that Pro²Assist is especially beneficial for users learning new procedural tasks, as it continuously provides guidance aligned with the task workflow. Regarding contextual relevance, 75% reported that the assistance closely matched their progress and ongoing actions. For timeliness, 40% rated the system as excellent and 55% as acceptable, indicating that the assistance is generally delivered at appropriate moments. Regarding intrusiveness, 55% found the system as minimally intrusive, while 20% felt that the assistance was somewhat lengthy or excessive. Notably, most of these 20% had prior task experience and agreed that the proactive timing and content were appropriate, but preferred more concise responses. For system preferences, 35% desired assistance delivery within 1 s, while the majority found longer latencies acceptable. Pro²Assist meets these expectations in most scenarios, achieving an average end-to-end latency within 5 s on an edge device and can be reduced to within 1 s on GPU platforms (details are in §5.6). For delivery modality, 60% preferred visual text overlays on the glasses display, consistent with Pro²Assist's current design. Another 30% preferred audio notifications, and 10% suggested adapting the delivery method to different task contexts. Note that while Pro²Assist primarily presents assistance through on-screen display, audio delivery is also supported through the smart glasses' built-in audio system.

Analysis of Negative Experiences. We further analyze negative experiences, which mainly arise from four aspects as follows.

- **Intrusiveness for experienced users.** While most participants did not find the responses intrusive, a few participants with prior task experience (P6, P13) preferred concise responses over detailed instructions. P6 mentioned, "Because I have done this task before, I just want the system to remind me of each step with concise guidance... it still feels intrusive to read through each time."

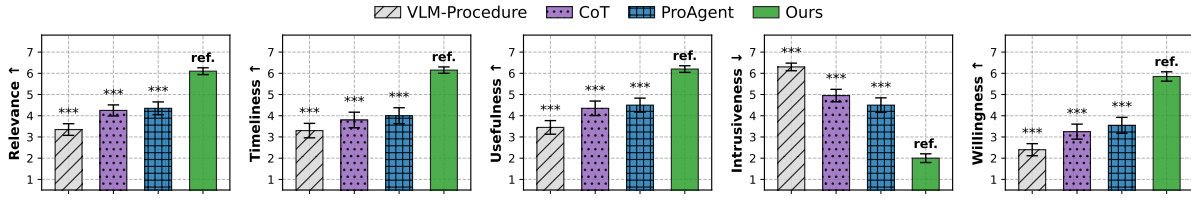


Fig. 32. Comparison of baselines and Pro²Assist using a 7-point Likert scale, ranging from 1 (Very low) to 7 (Very high). For *Intrusiveness*, lower is better. “ref.” marks reference, and *** $p < 0.001$ (paired Wilcoxon signed-rank test).

- *Unnecessary interruptions from false detections.* These typically occur in the middle of a step when the system is expected to remain silent. Some participants (P2, P5, P10) reported they noticed such interruptions but considered them acceptable, as they were mainly early reminders rather than unrelated guidance. For example, P10 mentioned, “*I was still scooping peanut butter and wanted more, but it said the step was done and to move on.*”
- *Mistimed assistance.* While 95% of participants rated *Timeliness* as excellent or acceptable, assistance can feel mistimed on short, familiar steps where users can move quickly from initiation to execution. P6, who had prior experience with the task, mentioned, “*I felt it was not useful when I needed to wait for the instruction on a short and simple step.*” This suggests timing tolerance depends on expertise, as experienced users can initiate actions quickly and have less tolerance for instructions that arrive after they are ready to execute.
- *Incorrect step guidance.* Such negative experiences arise from single-moment mispredictions, which are typically corrected by Pro²Assist’s subsequent predictions. While these errors might cause confusion, they were brief and resolvable in our study. This is reflected in *Relevance* ratings, where 90% of participants reported high or moderate alignment. Participants were able to handle such brief errors by relying on their own step awareness, informed by Pro²Assist’s prior correct step-aware guidance. P13 described, “*It first regarded my reach for cinnamon as reaching for banana slices next to it, but it corrected itself once I grabbed cinnamon.*” This highlights that quick recovery from mispredictions is critical to sustaining user experience.

Together with overall user perception, these findings show that Pro²Assist delivers contextually useful and timely assistance for most participants, while revealing design implications and opportunities for further improvement, which we discuss in §6.

User Experience Compared against Baselines. To further compare Pro²Assist with baselines from a user-experience perspective, we conducted a complementary study with the same 20 participants, who reviewed videos showing the proactive assistance generated by Pro²Assist and three representative baselines (VLM-Procedure, CoT, and ProAgent (FT)) and rated each system on a 7-point Likert scale across the five evaluation dimensions. For each participant, all four methods were applied to the same video to control for content effects, and the resulting outputs were presented in randomized order with method identities hidden to prevent bias. As shown in Figure 32, Pro²Assist consistently outperforms all three baselines across the five dimensions, with all gains being significant ($p < 0.001$). These subjective gains are consistent with the results in §5.3.1, indicating that Pro²Assist provides assistance that is more contextually aligned, better timed, more useful, and less intrusive, leading to higher willingness to use it.

6 DISCUSSION AND LIMITATIONS

In this section, we discuss the limitations and future directions of Pro²Assist, as well as design implications for proactive procedural assistants.

Sensing Modalities. Pro²Assist utilizes commonly available modalities on smart glasses, among which egocentric vision and head IMU provide important cues for understanding user actions and modeling user intent. Additional

sensing modalities available on emerging smart glasses (e.g., Meta Orion [6], Magic Leap 2 [5]) offer promising directions for future enhancement. For example, eye gaze tracking provides direct signals of user attention [65] and depth sensing provides explicit geometric and spatial cues [18, 34], which could enhance finer-grained intent modeling and richer hand-object spatial understanding.

VLM Reasoning and Enhancement. Our evaluation across base VLMs and ablation studies shows that while stronger VLMs improve Pro²Assist’s performance, our designed components remain essential for proactive procedural assistance, as they provide capabilities complementary to general VLM reasoning and enable system-level control over when to invoke inference and whether to deliver responses. In future work, Pro²Assist could further benefit from advances in LLM/VLM research beyond adopting stronger models. For instance, recent work [74] has shown the effectiveness of invoking external tools to provide assistance, which could be integrated into Pro²Assist to further enrich assistance. Additionally, advances in efficient VLM inference [61, 71, 77] provide potential to further reduce Pro²Assist’s latency on edge devices.

Error Detection in Procedural Tasks. Error detection is critical in procedural tasks [40, 60]. While Pro²Assist currently focuses on continuous step-aware guidance rather than explicit error detection, its multi-scale temporal context is promising for this extension, as hand-object interaction features [40] and comparing actions against expected steps [31, 60] are effective for procedural error detection.

Extension to Mixed-Initiative Assistance. While Pro²Assist delivers proactive assistance to help users perform tasks smoothly without explicit queries that may interrupt ongoing operations, reactive assistance [10, 35] provides complementary benefits by giving users direct control over when and what assistance to receive through on-demand interactions. Future work can extend Pro²Assist toward mixed-initiative assistance that combines the advantages of both proactive and reactive assistance modes, as responses to user queries can be directly grounded in the step-aware context continuously tracked by Pro²Assist.

Extension to Longitudinal Use. Pro²Assist currently focuses on continuous, single-session task execution. Extending to procedures spanning multiple days (e.g., sourdough baking) would require identifying procedural actions from continuous daily sensing and recovering task progress upon resumption. Additionally, leveraging cross-session experience over days for long-term personalization could enable adaptation to user-specific execution patterns for the procedural task. Pro²Assist’s multi-scale temporal context, which captures short-term hand manipulation cues and tracks per-session step progression, could be extended to support these scenarios. Recent advances in episodic memory [46, 63] and memory-augmented reasoning [24] offer further enabling techniques.

Design Implications. Our study reveals several design implications for future proactive procedural assistants. First, procedural assistance should minimize disruption to active task execution. Participants valued Pro²Assist’s tendency to remain silent during stable execution and deliver timely guidance at step transitions, since poorly timed interventions could break the user’s workflow. Second, assistants should recover from mispredictions to sustain user experience. Participants found brief errors acceptable when quickly resolved by subsequent responses, as they can handle such errors with their own task awareness informed by prior correct guidance. Third, assistance may need to be adaptive through user modeling. The study shows that the value of assistance varies with individuals, and participants less familiar with the task, who have the greatest need for guidance, benefit most from our system’s continuous step-aware guidance. Modeling user expertise and preferences would allow future systems to better adjust trigger timing and response verbosity.

7 CONCLUSION

This paper introduces Pro²Assist, an end-to-end system that provides continuous, step-aware guidance during procedural tasks. It leverages multimodal egocentric data from smart glasses and the reasoning capabilities of VLMs to continuously track, reason over, and assist the user’s evolving task state. Extensive evaluations show

that Pro²Assist significantly improves proactive procedural assistance, and a user study further confirms that users find it useful and are willing to adopt it in everyday procedural tasks.

Acknowledgments

This paper was supported in part by a Columbia Engineering Presidential Fellowship. This work was also partially supported by the Research Grants Council of Hong Kong under grants STG1/E-403/24-N and 14212323.

Use of Generative AI

Generative AI tools (OpenAI ChatGPT and Anthropic Claude) were used to improve the quality of writing, including style, phrasing, and grammar polishing of author-written text. Typical prompts included variations of “improve the writing of the following sentences to make it concise and clear without changing its meaning”. All AI-suggested edits were reviewed, verified, and revised by the authors.

References

- [1] 2025. Claude-Sonnet-4.5. <https://www.anthropic.com/claude/sonnet>.
- [2] 2025. Ollama. <https://ollama.com/>.
- [3] 2025. RayNeo X3 Pro. <https://rayneo.cn/x3pro.html>.
- [4] 2026. Google Gemini. <https://gemini.google.com/>.
- [5] 2026. Magic Leap 2 Devices. <https://www.magicleap.com/legal/devices-ml2>.
- [6] 2026. Meta Orion. <https://www.meta.com/emerging-tech/orion/>.
- [7] 2026. SentenceTransformers Documentation. <https://www.sbert.net/>.
- [8] Lavisha Aggarwal, Vikas Bahirwani, Lin Li, and Andrea Colaco. 2025. Generating Dialogues from Egocentric Instructional Videos for Task Assistance: Dataset, Method and Benchmark. *arXiv preprint arXiv:2508.11192* (2025).
- [9] Peri Akiva, Jing Huang, Kevin J Liang, Rama Kovvuri, Xingyu Chen, Matt Feiszli, Kristin Dana, and Tal Hassner. 2023. Self-supervised object detection from egocentric videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5225–5237.
- [10] Riku Arakawa, Jill Fain Lehman, and Mayank Goel. 2024. Prism-q&a: Step-aware voice assistant on a smartwatch enabled by multimodal procedure tracking and large language models. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 4 (2024), 1–26.
- [11] Riku Arakawa, Prasoon Patidar, Will Page, Jill Lehman, and Mayank Goel. 2025. Scaling Context-Aware Task Assistants that Learn from Demonstration and Adapt through Mixed-Initiative Dialogue. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–19.
- [12] Riku Arakawa, Hiromu Yakura, and Mayank Goel. 2024. PrISM-Observer: Intervention agent to help users perform everyday procedures sensed using a smartwatch. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–16.
- [13] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibao Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. 2025. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631* (2025).
- [14] Andrea Bandini and José Zariffa. 2020. Analysis of the hands in egocentric vision: A survey. *IEEE transactions on pattern analysis and machine intelligence* 45, 6 (2020), 6846–6866.
- [15] Siddhant Bansal, Chetan Arora, and C.V. Jawahar. 2022. My View is the Best View: Procedure Learning from Egocentric Videos. In *European Conference on Computer Vision (ECCV)*.
- [16] 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*. 12325–12341.
- [17] Justin Chan, Solomon Nsumba, Mitchell Wortsman, Achal Dave, Ludwig Schmidt, Shyamnath Gollakota, and Kelly Michaelson. 2024. Detecting clinical medication errors with AI enabled wearable cameras. *NPJ Digital Medicine* 7, 1 (2024), 287.
- [18] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. 2021. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9044–9053.

- [19] Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, YINUO Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*.
- [20] 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.
- [21] Tuochao Chen, Nicholas Scott Batchelder, Alisa Liu, Noah A Smith, and Shyamnath Gollakota. 2025. LLAMAPIE: Proactive In-Ear Conversation Assistants. In *Findings of the Association for Computational Linguistics: ACL 2025*. 13801–13824.
- [22] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24185–24198.
- [23] Sijie Cheng, Zhicheng Guo, Jingwen Wu, Kechen Fang, Peng Li, Huaping Liu, and Yang Liu. 2024. Egothink: Evaluating first-person perspective thinking capability of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14291–14302.
- [24] Dongwook Choi, Taeyoon Kwon, Dongil Yang, Hyojun Kim, and Jinyoung Yeo. 2025. Designing Memory-Augmented AR Agents for Spatiotemporal Reasoning in Personalized Task Assistance. In *2025 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 113–119.
- [25] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*. 1107–1128.
- [26] Anup Doshi and Mohan Manubhai Trivedi. 2009. On the roles of eye gaze and head dynamics in predicting driver’s intent to change lanes. *IEEE Transactions on Intelligent Transportation Systems* 10, 3 (2009), 453–462.
- [27] Anup Doshi and Mohan M Trivedi. 2012. Head and eye gaze dynamics during visual attention shifts in complex environments. *Journal of vision* 12, 2 (2012), 9–9.
- [28] Pardis Emami-Naeini, Janarth Dheenadhyalan, Yuvraj Agarwal, and Lorrie Faith Cranor. 2021. Which privacy and security attributes most impact consumers’ risk perception and willingness to purchase IoT devices?. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 519–536.
- [29] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. 2023. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561* (2023).
- [30] Alireza Fathi, Xiaofeng Ren, and James M Rehg. 2011. Learning to recognize objects in egocentric activities. In *CVPR 2011*. IEEE, 3281–3288.
- [31] Alessandro Flaborea, Guido Maria D’Amely Di Melendugno, Leonardo Plini, Luca Scofano, Edoardo De Matteis, Antonino Furnari, Giovanni Maria Farinella, and Fabio Galasso. 2024. Prego: online mistake detection in procedural egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18483–18492.
- [32] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2026. A survey on llm-as-a-judge. *The Innovation* 7, 6 (2026).
- [33] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- [34] Lin Huang, Boshen Zhang, Zhilin Guo, Yang Xiao, Zhiguo Cao, and Junsong Yuan. 2021. Survey on depth and RGB image-based 3D hand shape and pose estimation. *Virtual Reality & Intelligent Hardware* 3, 3 (2021), 207–234.
- [35] Yifei Huang, Jilan Xu, Baoqi Pei, Lijin Yang, Mingfang Zhang, Yuping He, Guo Chen, Xinyuan Chen, Yaohui Wang, Zheng Nie, et al. 2025. Vinci: A real-time smart assistant based on egocentric vision-language model for portable devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 9, 3 (2025), 1–33.
- [36] Youngkyoon Jang, Brian Sullivan, Casimir Ludwig, Iain Gilchrist, Dima Damen, and Walterio Mayol-Cuevas. 2019. EPIC-Tent: An Egocentric Video Dataset for Camping Tent Assembly. In *International Conference on Computer Vision (ICCV) Workshops*.
- [37] Glenn Jocher and Jing Qiu. 2024. *Ultralytics YOLO11*. <https://github.com/ultralytics/ultralytics>
- [38] Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305* (2018).
- [39] Anna Kukleva, Fadime Sener, Edoardo Remelli, Bugra Tekin, Eric Sauser, Bernt Schiele, and Shugao Ma. 2024. X-mic: Cross-modal instance conditioning for egocentric action generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 26364–26373.
- [40] Shih-Po Lee, Zijia Lu, Zekun Zhang, Minh Hoai, and Ehsan Elhamifar. 2024. Error detection in egocentric procedural task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18655–18666.

- [41] Chenyi Li, Guande Wu, Gromit Yeuk-Yin Chan, Dishita Gdi 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. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–24.
- [42] Yuanqi Li, Arthi Padmanabhan, Pengzhan Zhao, Yufei Wang, Guoqing Harry Xu, and Ravi Netravali. 2020. Reducto: On-camera filtering for resource-efficient real-time video analytics. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 359–376.
- [43] Yin Li, Zhefan Ye, and James M Reh. 2015. Delving into egocentric actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 287–295.
- [44] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs.CV] <https://arxiv.org/abs/1405.0312>
- [45] Jiacheng Liu, Yuanchun Li, Liangyan Li, Yi Sun, Hao Wen, Xiangyu Li, Yao Guo, and Yunxin Liu. 2024. ChainStream: An LLM-based Framework for Unified Synthetic Sensing. *arXiv preprint arXiv:2412.15240* (2024).
- [46] Yongdong Luo, Xianwu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, et al. 2024. Video-rag: Visually-aligned retrieval-augmented long video comprehension. *arXiv preprint arXiv:2411.13093* (2024).
- [47] Rachel McCloud, Carly Perez, Mesfin Awoke Bekalu, and K Viswanath. 2022. Using smart speaker technology for health and well-being in an older adult population: pre-post feasibility study. *JMIR aging* 5, 2 (2022), e33498.
- [48] Microsoft. 2026. Microsoft Azure Speech. Accessed: 2026-01-21. <https://azure.microsoft.com/en-us/products/ai-foundry/tools/speech>
- [49] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9879–9889.
- [50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
- [51] Kevin Pu, Ting Zhang, Naveen Sindhilnathan, Sebastian Freitag, Raj Sodhi, and Tanya R Jonker. 2025. ProMemAssist: Exploring Timely Proactive Assistance Through Working Memory Modeling in Multi-Modal Wearable Devices. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–19.
- [52] A Raouf and S Arora. 1980. Effect of informational load, index of difficulty direction and plane angles of discrete moves in a combined manual and decision task. *International Journal of Production Research* 18, 1 (1980), 117–128.
- [53] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems* 35 (2022), 25278–25294.
- [54] Yuhao Shen, Lu Wang, and Ehsan Elhamifar. 2021. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10156–10165.
- [55] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aidan Low, AJ Ostrow, Akhila Ananthram, et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267* (2025).
- [56] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. 2003. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 73–80.
- [57] Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*. Springer, 402–419.
- [58] Issatay Tokmurziyev, Miguel Altamirano Cabrera, Muhammad Haris Khan, Yara Mahmoud, and Dzmity Tsetserukou. 2026. LLM-glasses: GenAI-driven glasses with haptic feedback for navigation of visually impaired people. In *Companion Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*. 994–998.
- [59] Ashwin Vinod, Shrey Pandit, Aditya Vavre, and Linshen Liu. 2025. EgoVLM: Policy Optimization for Egocentric Video Understanding. *arXiv preprint arXiv:2506.03097* (2025).
- [60] Ruiqi Wang, Peiqi Gao, Patrick Lynch, Tingjun Liu, Yejin Lee, Carolyn Baum, Lisa Tabor Connor, and Chenyang Lu. 2025. CHEF-VL: Detecting Cognitive Sequencing Errors in Cooking with Vision-language Models. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 9, 4 (2025), 1–35.
- [61] Tiannan Wang, Wangchunshu Zhou, Yan Zeng, and Xinsong Zhang. 2023. Efficientvllm: Fast and accurate vision-language models via knowledge distillation and modal-adaptive pruning. In *Findings of the association for computational linguistics: ACL 2023*. 13899–13913.
- [62] Xijun Wang, Tanay Sharma, Achin Kulshrestha, Abhimitha Meka, Aavek Purohit, and Dinesh Manocha. 2025. EgoSocial: Benchmarking Proactive Intervention Ability of Omnimodal LLMs via Egocentric Social Interaction Perception. *arXiv preprint arXiv:2510.13105* (2025).
- [63] Ying Wang, Yanlai Yang, and Mengye Ren. 2023. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos. *arXiv preprint arXiv:2312.05269* (2023).
- [64] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

- [65] Ethan Wilson, Naveen Sendhilnathan, Charlie S Burlingham, Yusuf Mansour, Robert Cavin, Sai Deep Tetali, Ajoy Savio Fernandes, and Michael J Proulx. 2025. Eye gaze as a signal for conveying user attention in contextual AI systems. In *Proceedings of the 2025 symposium on eye tracking research and applications*. 1–7.
- [66] Jay Zhangjie Wu, David Junhao Zhang, Wynne Hsu, Mengmi Zhang, and Mike Zheng Shou. 2023. Label-efficient online continual object detection in streaming video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19246–19255.
- [67] Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. 2024. Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation. *Advances in Neural Information Processing Systems* 37 (2024), 109922–109947.
- [68] Zhenyu Xu, Hailin Xu, Zhouyang Lu, Yingying Zhao, Rui Zhu, Yujiang Wang, Mingzhi Dong, Yuhu Chang, Qin Lv, Robert P Dick, et al. 2024. Can large language models be good companions? An LLM-based eyewear system with conversational common ground. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 2 (2024), 1–41.
- [69] Jiaqi Yan, Ruilong Ren, Jingren Liu, Shuning Xu, Ling Wang, Yiheng Wang, Xinlin Zhong, Yun Wang, Long Zhang, Xiangyu Chen, et al. 2025. TeleEgo: Benchmarking Egocentric AI Assistants in the Wild. *arXiv preprint arXiv:2510.23981* (2025).
- [70] Bufang Yang, Yunqi Guo, Lilin Xu, Zhenyu Yan, Hongkai Chen, Guoliang Xing, and Xiaofan Jiang. 2025. Socialmind: Llm-based proactive ar social assistive system with human-like perception for in-situ live interactions. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 9, 1 (2025), 1–30.
- [71] Bufang Yang, Lixing He, Neiwen Ling, Zhenyu Yan, Guoliang Xing, Xian Shuai, Xiaozhe Ren, and Xin Jiang. 2023. Edgefm: Leveraging foundation model for open-set learning on the edge. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*. 111–124.
- [72] Bufang Yang, Lixing He, Kaiwei Liu, and Zhenyu Yan. 2024. Viassist: Adapting multi-modal large language models for users with visual impairments. In *2024 IEEE International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things (FMSys)*. IEEE, 32–37.
- [73] Bufang Yang, Lilin Xu, Liekang Zeng, Yunqi Guo, Siyang Jiang, Wenrui Lu, Kaiwei Liu, Hancheng Xiang, Xiaofan Jiang, Guoliang Xing, et al. 2025. ProAgent: Harnessing On-Demand Sensory Contexts for Proactive LLM Agent Systems. *arXiv preprint arXiv:2512.06721* (2025).
- [74] Bufang Yang, Lilin Xu, Liekang Zeng, Kaiwei Liu, Siyang Jiang, Wenrui Lu, Hongkai Chen, Xiaofan Jiang, Guoliang Xing, and Zhenyu Yan. 2025. ContextAgent: Context-Aware Proactive LLM Agents with Open-World Sensory Perceptions. *arXiv preprint arXiv:2505.14668* (2025).
- [75] Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. 2025. Egolife: Towards egocentric life assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 28885–28900.
- [76] Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, et al. 2025. MMEgo: Towards building egocentric multimodal LLMs for video QA. In *International Conference on Learning Representations*.
- [77] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. 2024. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417* (2024).
- [78] Shaoze Zhou, Diana Nelly Rivera Rodriguez, Pedro Remior, Joaquin Frangi, Lingyao Li, Renkai Ma, Janet G Johnson, Christine Lisetti, and Chen Chen. 2026. Exploring Needs and Design Opportunities for Proactive Information Support in In-Person Small-Group Conversations. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*. 1–7.
- [79] Sheng Zhou, Junbin Xiao, Qingyun Li, Yicong Li, Xun Yang, Dan Guo, Meng Wang, Tat-Seng Chua, and Angela Yao. 2025. Egotextvqa: Towards egocentric scene-text aware video question answering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 3363–3373.
- [80] Chenfei Zhu, Shao-Kang Hsia, Xiyun Hu, Ziyi Liu, Jingyu Shi, and Karthik Ramani. 2025. agentAR: Creating Augmented Reality Applications with Tool-Augmented LLM-based Autonomous Agents. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–23.

A APPENDIX

A.1 Detailed Annotation Procedure

As shown in Figure 15, the annotation pipeline consists of a two-stage procedure that combines human expertise with LLM-assisted generation. In the first stage, human annotators are required to mark time interval boundaries of fine-grained step execution status S_s (i.e., *just start* (at the beginning of the step), *in progress* (in the middle of the step), *about to finish* (near completion of the step), and *step transition* (between two steps where the user transitions from one to the next)) based on the step description D_s and observable hand movements. Based on the annotated S_s and D_s , the historical task progress H_p is constructed to summarize the completed steps and task progress over time. Annotators also label the proactive trigger P_t , indicating whether proactive assistance is required. To ensure quality and consistency, a cross-validation process is adopted in which annotators review each other's annotations. Additionally, to avoid redundancy from visually similar frames, annotators select diverse and representative moments and filter out highly similar samples.

In the second stage, advanced LLMs are employed to generate motion-aware action understanding A_u and step-aware proactive responses P_r . For motion-aware action understanding, the LLM is prompted with the step description D_s , hand motion cues M_h , and human-annotated step status S_s , together with the visual input. The hand motion cues are extracted from each pair of consecutive frames (I_{t-1}, I_t) as described in §4.3.2. The LLM then produces detailed action understanding annotations describing the user's current state. For proactive response generation, the prompt includes the historical task progress H_p , the generated action understanding A_u , and the structured procedural guideline $\mathcal{G}$ (constructed as described in §4.3.1), together with the visual input. The LLM generates contextually appropriate proactive responses P_r aligned with the user's current state. Finally, all LLM-generated annotations (A_u, P_r) are verified by two human annotators to ensure accuracy, relevance, and appropriateness. The prompt templates used for LLM-based annotation generation are shown in Figure 34 and 35.

A.2 Prompts

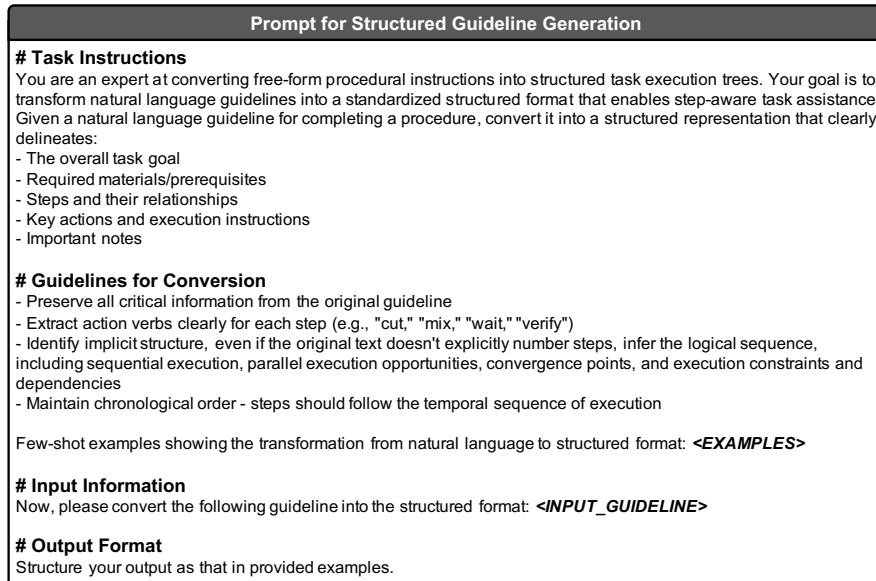


Fig. 33. Prompt template for structured guideline generation.

Prompt for Motion-Aware Action Understanding Generation
Task Instructions You are an expert annotator for egocentric procedural tasks. Your goal is to generate a precise Motion-Aware Action Understanding description that helps models correctly determine the step identity and the step progress status. You will be given: - Image: the raw egocentric scene - The ground truth step name - The ground truth step status - Motion cues describing the hand movement trend extracted from optical flow. You may combine the image-plane hand motion cue with visual context, object relationships, and task semantics to infer how the hand is moving in the actual scene when useful. Please carefully integrate the visual evidence from Image, the motion cue text and provided step and step status. Provide a concise but informative Action Understanding description that: - Describes what the hand is doing - Mentions relevant objects if visible - Reflects how the motion relates to this procedural step - Justifies why this corresponds to the given step status - Is written as natural language (not bullet points, not mechanical) Do NOT simply restate the step or status. Instead, explain what the user is doing in a way that supports why this step and status make sense. # Input Information [Image]: <IMAGE> [Motion Cue]: The motion between the previous and current frame indicates that: <MOTION_CUES> [Ground Truth Step]: <STEP> [Ground Truth Step Status]: <STATUS> # Output Format <1–2 sentences of clear reasoning-based description>

Fig. 34. Prompt template for motion-aware action understanding generation.

Prompt for Step-Aware Proactive Response Generation
Task Instructions You are an expert annotator for procedural tasks. Analyze the current frame and generate a Thinking Trace, Response, and Possible Next Step based on the provided context. You will be given: - Image: the raw egocentric scene - The guideline containing task-specific expert knowledge - Historical task progress indicating the completed steps and their sequences - Motion-aware action understanding about this frame Generate a brief (less than 50 words) and helpful message based on the input contexts by integrating the user's current state and procedural expert knowledge to guide the user perform the procedural task smoothly. Please provide relevant guidance for the current step when the step is just start; suggest possible next steps when the step is about to finish, and make a recommendation and provide reasoning if multiple valid next steps exist (parallel branches). # Input Information [Image]: <IMAGE> [Overall Task Guideline]: <GUIDELINE> [Task Progress]: <COMPLETED_STEPS> [Action Understanding]: <ACTION_UNDERSTANDING> # Output Format <Your response >

Fig. 35. Prompt template for step-aware proactive response generation.

Prompt for LLM-as-a-Judge Evaluation
Task Instructions You are an expert evaluator of proactive assistants operating in multi-step procedural tasks. You must judge whether a model's proactive response is appropriate given the current step, execution status, guideline, and completed steps. # Evaluation Criteria 1. Relevance (0-5 points) Evaluate how well the proactive response adheres to the current context and task guidelines. - 5: Perfectly matches the current step, status and guideline constraints - 4: Mostly correct, minor redundancy or slight mismatch - 3: Partially relevant but unclear, premature, or incomplete - 2: Weakly related to the current step - 1: Mostly wrong step or wrong timing - 0: Completely unrelated or harmful 2. Usefulness (0-5 points) Evaluate how helpful and comprehensible the response is for the user in the current context. - 5: Extremely useful, provides clear actionable guidance, anticipates user needs perfectly - 4: Very useful, clear and helpful with minor room for improvement - 3: Moderately useful, provides some value but could be clearer or more actionable - 2: Somewhat useful, contains helpful elements but is confusing or incomplete - 1: Minimally useful, very limited practical value or clarity - 0: Not useful, confusing, misleading, or provides no value to the user # Input Information Ground Truth Data: - Current Step: <STEP> - Step Status: <STATUS> - Overall Task Guideline: <GUIDELINE> - Task Progress: <COMPLETED_STEPS> Predicted Proactive Response: <PROACTIVE RESPONSE> # Output Format Provide your evaluation in the following JSON format: Relevance Score: [0-5] Usefulness Score: [0-5]

Fig. 36. Prompt template for LLM-as-a-Judge Evaluation.