Proactive Assistant Dialogue Generation from Streaming Egocentric Videos

Yichi Zhang[1,2,*], Xin Luna Dong[1], Zhaojiang Lin[1], Andrea Madotto[1],
Anuj Kumar[1], Babak Damavandi[1], Joyce Chai[2], Seungwhan Moon[1]

[1]Meta   [2]University of Michigan

Correspondence: zhangyic@umich.edu

Abstract

Recent advances in conversational AI have been substantial, but developing real-time systems for perceptual task guidance remains challenging. These systems must provide interactive, proactive assistance based on streaming visual inputs, yet their development is constrained by the costly and labor-intensive process of data collection and system evaluation. To address these limitations, we present a comprehensive framework with three key contributions. First, we introduce a novel data curation pipeline that synthesizes dialogues from annotated egocentric videos, resulting in PROASSIST, a large-scale synthetic dialogue dataset spanning multiple domains. Second, we develop a suite of automatic evaluation metrics, validated through extensive human studies. Third, we propose an end-to-end model that processes streaming video inputs to generate contextually appropriate responses, incorporating novel techniques for handling data imbalance and long-duration videos. This work lays the foundation for developing real-time, proactive AI assistants capable of guiding users through diverse tasks. Project page: https://pro-assist.github.io/

1 Introduction

Recent advances in multimodal language models have transformed various aspects of human-AI interaction (Achiam et al., 2023; Team et al., 2024; Dubey et al., 2024). However, developing AI systems capable of providing real-time, interactive guidance for physical tasks remains a significant challenge (Bao et al., 2023). Ideally, as illustrated in Figure 1, such an assistant should proactively guide users through each step of a task based on a high-level goal, determining both when and how to communicate through continuous processing of the environment and understanding of task objectives. This requires the system to handle streaming video

---

Figure 1: An example conversation between user and their task assistant. The assistant receives real-time video streams from the user's perspective, and provides proactive guidance to assist with the task. Images excerpted from Ego4D (Grauman et al., 2022).

inputs while simultaneously managing diverse user interactions including requests, questions, and comments, and offering timely guidance upon detecting the completion of each task step. The dual challenge of determining both appropriate response timing and content through real-time processing of long-horizon video inputs makes task guidance a particularly complex problem.

Despite significant progress in component technology such as action recognition (Damen et al., 2020; Grauman et al., 2022), mistake detection (Sener et al., 2022; Wang et al., 2023b; Lee et al., 2024), and question answering (Wong et al., 2022; Ilaslan et al., 2023), we are still far from enabling the holistic ability to generate appropriate dialogue responses for task guidance. Two major challenges have hindered progress toward this goal: the lack of large-scale and diverse training data, as existing datasets are primarily constrained by labor-intensive Wizard-of-Oz setups limited to single domains (Bao et al., 2023; Wang et al., 2023b); and the lack of scalable evaluation frameworks for

assistant dialogue generation that could serve as efficient proxies for human evaluation to support rapid and reproducible model comparisons during system development.

To address these limitations, we propose a new problem of proactive assistant dialogue generation from streaming videos and develop comprehensive resources to approach this problem. We introduce an automated approach for synthesizing task-oriented dialogues from well-annotated egocentric video datasets (Song et al., 2024b; Huang et al., 2024; Grauman et al., 2024; Damen et al., 2020; Sener et al., 2022), resulting in PROASSIST– a large-scale synthetic dialogue dataset containing 30,135 dialogues across 479 hours of video in cooking, object manipulation, assembly, and laboratory domains. Our method leverages state-of-the-art large language models (Achiam et al., 2023; Anthropic, 2024; Dubey et al., 2024) to generate realistic assistant-user interactions, using detailed timestamped video descriptions to maintain temporal alignment. For systematic evaluation, we propose two complementary automatic metrics: a pairwise approach based on sentence matching and an end-to-end approach utilizing LLM-as-a-Judge (Zheng et al., 2023). Through extensive human studies, we validate both the quality of our synthetic data and the alignment between our proposed metrics and human judgment, establishing PROASSIST as a reliable benchmarking resource.

Based on PROASSIST, we develop an end-to-end multimodal large language model (MLLM) for generating contextually appropriate responses from streaming video inputs. Building upon the VideoLLM-Online architecture (Chen et al., 2024), we introduce two key innovations: negative frame sub-sampling to improve response timing decisions, and iterative progress summarization to enable efficient processing of long video sequences. Our experimental results demonstrate the effectiveness of these modeling techniques while providing valuable insights into the complexities of perceptual task guidance.

2   Related Work

Interactive Assistant for Task Guidance.   Task guidance systems have evolved from early rule-based policies (Ockerman and Pritchett, 1998, 2000) to perception-enabled but task-specific solutions (Leelasawassuk et al., 2017; Reyes et al., 2020; Lu and Mayol-Cuevas, 2019; Wang et al., 2016; Sato et al., 2014; Bao et al., 2023). Re-

cent research has primarily focused on developing components for single-domain systems (Wang et al., 2023b), including environment understanding (Wong et al., 2022; Ilaslan et al., 2023), user behavior analysis (Damen et al., 2020; Grauman et al., 2022; Huang et al., 2024), and mistake detection (Sener et al., 2022; Lee et al., 2024; Peddi et al., 2023). Our work differs by evaluating end-to-end dialogue generation capabilities of a general-purpose system across multiple domains.

Synthetic Dialogue Generation.   Synthetic dialogue generation has proven effective for creating large-scale datasets in both text-only (Shah et al., 2018; Mohapatra et al., 2021; Rastogi et al., 2020) and multimodal scenarios (Kottur et al., 2021; Wu et al., 2023; Zhan et al., 2024; Moon et al., 2020). Recent LLMs have shown remarkable capabilities in simulating human behavior (Park et al., 2023), particularly valuable for low-resource scenarios (Li et al., 2022; Abdullin et al., 2024; Chen et al., 2023). While previous work has demonstrated the feasibility of generating dialogues about visual content using structured descriptions (Liu et al., 2024a; Maaz et al., 2023; Luo et al., 2023; Chen et al., 2024), our approach uniquely employs a dedicated LLM pipeline to generate natural assistant-user interactions for egocentric task completion videos.

Multimodal Dialogue Modeling.   The success of large language models (LLMs) (Brown, 2020; Ouyang et al., 2022) has led to the development of multimodal variants (MLLMs) capable of handling both image-based (Alayrac et al., 2022; Liu et al., 2024a; Li et al., 2023a; Zhu et al., 2023b) and video-based dialogues (Li et al., 2023b; Lin et al., 2023; Maaz et al., 2023; Song et al., 2024a; Yang et al., 2023; Zhang et al., 2023). However, these models typically operate in an offline setting with access to complete videos. While VideoLLM-Online (Chen et al., 2024) represents a significant step toward online video processing, it primarily focuses on short video clips. Our work extends the model with novel techniques specifically designed for task guidance scenarios, introducing mechanisms for response timing decisions and efficient processing of long-horizon videos.

3   Proactive Assistant Dialogue Generation from Streaming Videos

3.1   Problem Definition

The goal of a proactive assistant system is to generate prompt, appropriate, and helpful guidance from

| Task: Prepare Seafood Stir-Fry |
| Recipe |
| 1. Prepare ingredients: wash and cut fish, crab stick, and sausage. 2. Prepare vegetables: wash and cut. 3. Heat oil in a pan and add ingredients: fish, crab stick, sausage, and vegetables. 4. Stir-fry the ingredients 5. Adjust heat and conti-nue cooking until done. 6. Season and serve. |

[1.5s] User: Hi, I want to prepare a seafood stir-fry.

[2.0s] Assistant: Great choice! To start, let's prepare the cutting board. Can you take it out and wash it?

[9.5s] Assistant: The sink looks full. You can put the pot on the cooker.

[25.5s] User: How do I wash it?

[25.5s] Assistant: Just rinse it with some water and dry it with a towel. We want it clean and dry for cutting.

[33.5s] Assistant: Now, take out the knife and wash it as well. Make sure it's clean and dry before we start cutting.

[48.0s] Assistant: Let's get the ingredients. Can you open the refrigerator and take out the egg, crab stick, and fish? Take out the egg first, we won't be using it for a bit.

[75.5s] User: How do I prepare the crab stick

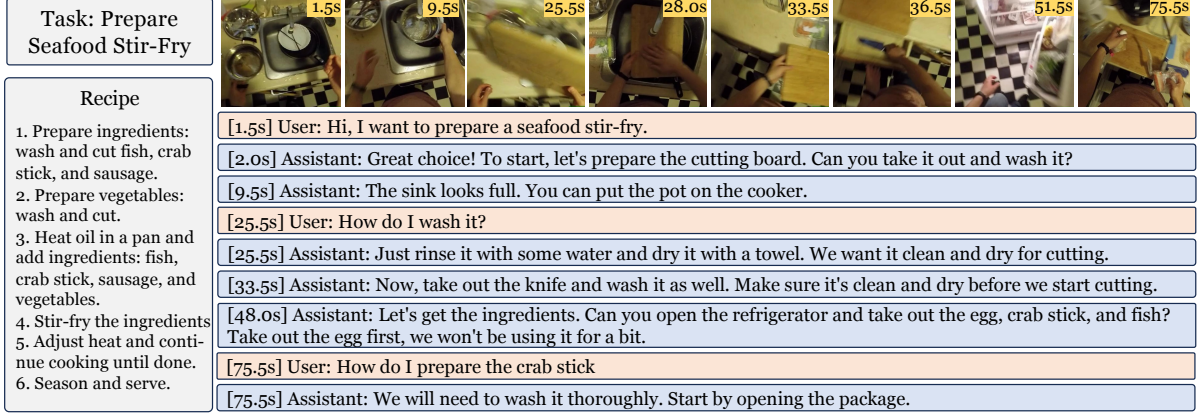[75.5s] Assistant: We will need to wash it thoroughly. Start by opening the package.

Figure 2: A cooking task example from PROASSIST, derived from EpicKitchen. The task goal, recpie and dialogue are generated through our synthetic data curation pipeline (dialogue partially shown due to space constraints).

egocentric video streams in real time. We formu-late this problem as a streaming video-to-dialogue generation task. Given a video stream of T frames, our objective is to generate a sequence of assis-tant responses $s_{1:T}$ that maximizes the conditional probability:

$$s_{1:T} = \arg\max \prod_t P(s_t|v_{1:t}, s_{1:t-1}, k) \quad (1)$$

where s_t represents the assistant's response at time step t (either a textual message to user or $\varnothing$ for keeping silence), v_t denotes the multimodal input including the video frame and optional user utter-ance input, and k represents optional task knowl-edge (e.g., a recipe). The task begins when the user provides a goal through text input. When k is pro-vided, we term this knowledge-conditioned evalua-tion, reflecting a realistic retrieval-augmented setup for real-world systems. This formulation requires the assistant to determine both when to speak and what to say based on current visual context, dia-logue history, and task understanding.

3.2 PROASSIST: A Synthetic Dialogue Dataset
We now describe our approach for the creation of PROASSIST. We first collect egocentric videos that are extensively annotated with timestamped user action descriptions from six public dataset: Ego4D-Goalstep (Grauman et al., 2022; Song et al., 2024b), EpicKitchen (Damen et al., 2020), HoloAssist (Wang et al., 2023b), Assembly101 (Sener et al., 2022), EgoExoLearn (Huang et al., 2024), and WTaG (Bao et al., 2023). The anno-tations are processed into a standardized format, [t]<description>, where [t] represents a times-tamp or time span. Additional annotations such as high-level task step and error correction labels are similarly formatted and inserted in chronolog-ical order whenever available. This unified repre-sentation enables LLMs to effectively understand

ongoing activities at each time step.

Building on these annotations, we design a data curation pipeline consisting of the following steps:

1. Task Goal and Recipe Generation: We first prompt the LLM to summarize the task goal and generate a task recipe based on the video descrip-tions. This step will be skipped if the dataset already includes these elements (e.g., WTaG). The generated goal serves as the initial user in-put describing the task, while the recipe supports knowledge-conditioned evaluation (§6.1).

2. Video Pre-Filtering: Non-procedural, multi-tasking, or incompletely annotated videos are fil-tered out to ensure dataset quality.

3. Multi-Round Dialogue Generation: Dialogues are generated using three types of user behavior: *no talk* (i.e., silent except for giving the goal), *talk some* (occasional task-related questions), and *talk more* (frequent conversational interactions). Inputs include the goal, video descriptions, and user be-havior type. To handle long video descriptions, we adopt a multi-round generation approach, dividing videos into chunks and generating dialogues incre-mentally to stay within the LLM's context window. Afterward, we prompt the LLM for a refinement pass to improve dialogue naturalness and coher-ence. At this step, we generate 10 dialogues per video, distributed across user types in a 2:4:4 ratio.

4. Dialogue Annotation: The generated dialogues are then labeled by LLM, including assistant in-tent (instruction, mistake correction, feedback) and response type (responsive or proactive). Addition-ally, we also generate a summary of progress at each assistant turn for the user's progress so far, which will be used to support the iterative progress summarization approach (§5.3).

5. Automatic Quality Evaluation and Post-

Filtering: We perform automatic evaluations to ensure the dialogues meet high standards, assessing timing precision, task step coverage, and assistant responsiveness. Low-quality dialogues are filtered out from the training set. For the validation split, we retain only the highest-scoring dialogue per user type, splitting them evenly into validation and test sets. This process removes approximately 25% of dialogues and 41 hours of video. Final data statistics are shown in Table 1.

We leverage LLaMA-3.1-70B-Instruct (Dubey et al., 2024) as the LLM for all the aforementioned steps. An example dialogue generated through this pipeline is shown in Figure 2. To ensure the safety of our generated dataset, we applied the LLaMA-Guard-3-8B model[1] to all generated dialogues for safety classfication. The classifier flagged 17 instances (0.05%) as potentially unsafe. Upon manual inspection, we found no actual issues in these flagged cases, indicating that they were likely false positives. More details including the prompt for each step, data distributions and dialogue examples are available in the Appendix.

Subset	Video Hour	#Videos	#Dialogues
Ego4D	136.6 / 11.6 / 13.2	382 / 32 / 33	3182 / 96 / 99
HoloAssist	107.0 / 7.6 / 6.8	1436 / 97 / 97	7052 / 291 / 291
EgoExoLearn	68.5 / 8.8 / 9.4	321 / 41 / 41	3210 / 123 / 123
Assembly101	43.1 / 6.9 /7.2	756 / 112 / 112	7492 / 336 / 336
EpicKitchens	34.0 / 4.2 / 4.1	320 / 50 / 50	6376 / 150 / 150
WTaG	7.1 / 1.2 / 1.3	40 / 7 / 7	786 / 21 / 21
Total	478.7	3934	30135

Table 1: Data statistics of PROASSIST for train/validation/test splits. More statistics in Appendix A.3.

4 Evaluation of Proactive Task Assistant

Evaluating interactive dialogue systems is inherently challenging (Deriu et al., 2021), particularly for proactive task guidance where both response timing and content must be assessed. While direct human evaluation through system interaction would be ideal, the high cost makes it impractical for large-scale benchmarking, especially during rapid development cycles. We therefore propose an offline evaluation framework that enables efficient, automatic dataset-based assessment.

Our framework aims to measure the overall helpfulness of a system's sequential predictions $s_{1:T}$ for a video stream $v_{1:T}$ (as defined in Eq.1) by comparing them against ground-truth dialogue $\hat{s}_{1:T}$. This comparison is challenging due to the potential long task horizon and the need to evaluate predictions

that may differ from ground-truth in both timing and content. Below, we introduce two evaluation metrics designed to address these challenges.

Pairwise Evaluation via Sentence Matching
Our first metric evaluates system performance by matching each predicted utterance with semantically similar and temporally aligned reference utterances. Based on these matches, we compute three metrics: *precision* (matched predictions over total predictions), *recall* (matched predictions over total references), and *F1* (their harmonic mean). To identify optimal matches, we apply bipartite matching based on a cost matrix combining both semantic and temporal alignment costs. The semantic cost between predictions s_i and references $\hat{s}_j$ is defined as:

$$s(i,j) = \begin{cases} 1 & \text{if } \hat{s}_j = \varnothing \\ 1 - sim(e(s_i), e(\hat{s}_j)) & \text{else} \end{cases}$$

where $e(\cdot)$ is the sentence embedding function and $sim(\cdot)$ denotes cosine similarity. The temporal cost encourages matching with temporally proximate messages:

$$d(i,j) = \begin{cases} |i-j|^p & \text{if } i-j \in [-L, R] \\ \infty & \text{else} \end{cases}$$

where p controls the cost increase rate with time difference, while R and L define maximum allowable time differences for predictions preceding or following references. We set $R < L$ to favor earlier predictions, preventing the model from exploiting future frame information. The final matches are computed using the LAPJVsp algorithm (Jonker and Volgenant, 1988) with a weighted sum of both costs. More details are provided in the Appendix.

End-to-End Evaluation via LLM-as-a-Judge
Our ultimate goal is to evaluate the assistant's overall usefulness to the user. While the pairwise matching approach approximates this through prediction-reference similarity, it cannot capture the flexibility of different guidance strategies. Drawing inspiration from recent LLM-based evaluation approaches (Zheng et al., 2023; Liu et al., 2023; Maaz et al., 2023), we propose using LLMs to directly assess the quality of the overall assistance experience. Given timestamped predictions and reference dialogues, we prompt an LLM to evaluate system performance across four dimensions: *correctness* of guidance and feedback, *appropriateness* of response timing, *efficiency* of information delivery, and *overall* helpfulness. Each aspect is rated on a 5-point Likert scale from "very poor" to "excellent". For reliability, we average scores from three
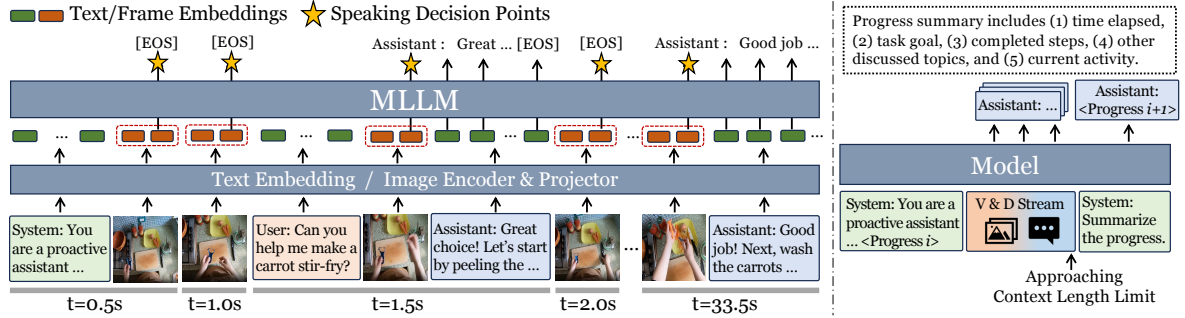
Figure 3: **Left**: Streaming video-to-dialogue generation using VideoLLM-Online. The model processes live video frames and optional textual inputs to decide whether to speak or remain silent at designated decision points (yellow stars), and autogressively generates assistant responses as needed. Learning when to speak faces significant class imbalance due to the sparsity of speaking frames. **Right**: Illustration of iterative progress summarization. When approaching its context length limit, the model generates a concise task progress summary, then restarts generation with this summary incorporated into a new system prompt.

independent runs. The complete evaluation prompt is provided in the Appendix.

Note that our metrics are generally applicable beyond PROASSIST to any dataset with ground-truth dialogues. Moreover, the pairwise evaluation can be applied to other streaming video-to-text tasks requiring joint assessment of timing and content, such as online action narration (see §6.1).

5 Proactive Assistant Dialogue Modeling

Next, we present our exploration into developing a functional proactive assistant dialogue generation model. We begin with an analysis of existing models to assess their feasibility in addressing our problem. Then, we describe how we enhance a baseline model with two novel techniques to enable it to tackle the unique challenges involved.

5.1 Feasibility Analysis of Existing Models

Streaming video-to-dialogue generation poses unique modeling challenges to real-time video processing and online text generation. Most existing MLLMs (Lin et al., 2023; Maaz et al., 2023; Zhang et al., 2023; Li et al., 2025; Moon et al., 2024; He et al., 2024; Weng et al., 2025; Wang et al., 2024) are designed for offline scenarios where the complete video is available beforehand, making them unsuitable for our setup. While state-of-the-art proprietary MLLMs (Achiam et al., 2023; Anthropic, 2024; Team et al., 2024) can process interleaved image-text inputs, they suffer from high API latency and cost[2] and often struggle to determine appropriate response timing (Chen et al., 2024).

VideoLLM-Online (Chen et al., 2024) offers a viable baseline for our task, as it specifically handles

---

[2]For example, deciding when and what to say at 2 FPS for a 30-minute video requires making 6000 API calls.

live-streamed video inputs. As shown in Figure 3, the model processes interleaved video frames and textual inputs by encoding frames into visual tokens through a frozen pretrained image encoder and a tunable projector layer. For each frame, it predicts whether to respond at the last visual token position, generating [EOS] to remain silent or initiating response generation otherwise. To enable real-time interaction, it employs a compact frame representation of 1-10 tokens, significantly fewer than mainstream MLLMs.

However, VideoLLM-Online faces two key limitations in our task guidance scenario: the difficulty of learning when to speak due to the sparsity of speaking moments, and the inability to handle long-horizon tasks due to context window constraints. In the following sections, we present our enhanced model that addresses these challenges through two novel techniques.

5.2 Learning When to Speak under Imbalance

Learning when to speak can be framed as a sequence of binary decisions, where at each step the model must choose between speaking and remaining silent. We denote frames requiring responses as positive samples and those requiring silence as negative samples. As shown in Figure 3 (left), the speaking decision points (yellow stars) demonstrate a significant imbalance, with far more negative samples (predicting [EOS]) than positive ones (predicting Assistant). This imbalance creates a challenging learning problem, as directly optimizing cross-entropy on the original distribution leads to a classifier biased toward silence.

We propose Negative Frame Sub-sampling (NFS) to address this challenge. During training, we compute gradients only for positive frames and

5

a uniformly sampled subset of negative frames, comprising a proportion ρ of total negative samples. The loss remains unchanged for non-decision positions to maintain response generation capability. This approach can be efficiently implemented by adjusting the gradient computation mask without modifying model inputs. Furthermore, dynamically resampling negative samples each epoch ensures all positions can potentially contribute to learning over time, enhancing model robustness.

5.3 Iterative Progress Summarization

Long-horizon tasks (e.g., hour-long videos) challenge models in tracking goals and progress over time during both training and inference. Hardware constraints (e.g., GPU memory) during training enforce fixed-length sequence processing (L), forcing truncation of longer samples[3], causing substantial information loss and hindering the learning of long-horizon task progressions. During inference, context length limitations similarly restrict processing to tasks within the model's training window.

We introduce Iterative Progress Summarization (IPS) to overcome these issues, enabling continuous task tracking via dynamic memory compression. As shown in Figure 3 (right), when approaching context limits, the model generates a concise, task-relevant progress summary. Generation then resumes with this summary incorporated into the initial system prompt for the next processing segment. In training, long videos are preprocessed into context-fitting chunks with summaries carried forward. Critically, unlike methods requiring specialized training (Wang et al., 2023a; Chevalier et al., 2023), IPS integrates with standard LLM training, enabling our model to handle potentially infinite-length video streams while maintaining task and progress tracking.

6 Experiments

6.1 Experiment Setups

Baseline Task. While we primarily evaluate our model on proactive assistant dialogue generation, we also include egocentric action narration as a baseline task, where the model describes the camera wearer's actions in real-time. Action narration serves as a simpler variant of streaming video-to-text generation that mainly requires visual perception capabilities. By comparing performance between action narration and dialogue generation, we
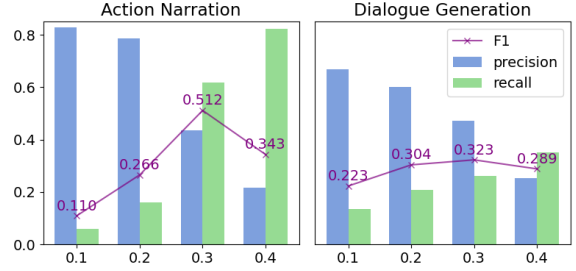

Figure 4: Model performance under different speaking decision threshold. The trade-off between precision and recall exists across both tasks.

can better understand how well our model handles capabilities beyond visual perception, such as situational reasoning and progress tracking, which are essential for effective task guidance.

Knowledge-Conditioned Evaluation. We introduce a knowledge-conditioned evaluation setup where the model receives task-specific instructions (e.g., recipes) as a system prompt after the user states their goal. This setup mirrors real-world scenarios where assistants access user-provided recipes or retrieved knowledge to offer guidance.

Speaking Decision Threshold. At inference time, we convert the model's probabilistic token predictions into binary decisions using a threshold θ: the model remains silent if the probability of [EOS] exceeds θ. Our experiments show that model performance is highly sensitive to θ, with a clear precision-recall tradeoff (Figure 4). We use the θ with the highest validation F1 score for testing.

Model Variants. We implement three variants of VideoLLM-Online, with different number of visual tokens per frame: $I = 1, 5, 10$. The model is intuitively better at visual perception with more tokens, with a cost of computationally more expensive.

Implementation Details. We use LLaMA-3.1-8B-Instruct (Dubey et al., 2024) as the backbone and SigLIP-SO400M (Zhai et al., 2023) as the frame encoder for VideoLLM-Online. For training, we adopt a single stage training on mixed data of dialogues both with and without knowledge from PROASSIST, Ego4D online action narration, and some auxiliary vision-language datasets, resulting in a single model that can be tested with different setup and tasks. See the Appendix for more details.

6.2 Dialogue Quality of PROASSIST

To validate PROASSIST as a reliable resource for studying proactive assistant dialogue generation, we conducted a comprehensive human evaluation of the synthetic dialogues. We uniformly sampled 100 dialogues from the test split across all six data subsets, covering three user types (i.e., *no*

---

[3]For example, 86% of samples from the Ego4D subset of PROASSIST must be truncated when L=4096 and each frame is encoded as 10 tokens.

	Correctness	Helpfulness	Alignment	Naturalness
All	$3.27_{\pm 0.79}$	$3.46_{\pm 0.77}$	$2.91_{\pm 1.00}$	$3.54_{\pm 0.70}$
- No Talk	$3.27_{\pm 0.70}$	$3.47_{\pm 0.74}$	$2.75_{\pm 0.96}$	$3.32_{\pm 0.79}$
- Talk Some	$3.23_{\pm 0.86}$	$3.37_{\pm 0.80}$	$2.93_{\pm 1.01}$	$3.50_{\pm 0.74}$
- Talk More	$3.32_{\pm 0.79}$	$3.53_{\pm 0.76}$	$3.05_{\pm 0.99}$	$3.80_{\pm 0.44}$
HoloAssist-Gen	$3.15_{\pm 0.91}$	$3.40_{\pm 0.49}$	$2.65_{\pm 0.91}$	$3.60_{\pm 0.73}$
HoloAssist-Human	$2.88_{\pm 1.05}$	$2.62_{\pm 1.11}$	$2.75_{\pm 1.09}$	$2.50_{\pm 1.32}$
WTaG-Gen	$3.50_{\pm 0.59}$	$3.50_{\pm 0.92}$	$3.15_{\pm 1.11}$	$3.65_{\pm 0.73}$
WTaG-Human	$3.60_{\pm 0.49}$	$3.60_{\pm 0.66}$	$3.60_{\pm 0.66}$	$3.60_{\pm 0.66}$

Table 2: Human evaluation of the generated dialogue quality. For HoloAssist and WTaG where human-collected dialogues are available, we evaluate them using the same approach for a side-by-side comparison with our generated dialogues.

Metric	P	S
F1 vs Human	0.35^{**}	0.32^{*}
Overall vs Human	0.47^{**}	0.44^{**}
Overall vs F1	0.67^{**}	0.64^{**}

Table 3: Pearson and Spearman coefficient between our metrics and human judgment (*: $p < 0.05$, **: $p < 0.01$).

Metric	A.N.	D.G.
F1	0.80	0.67
Precision	0.53	0.42
Recall	0.47	0.63

Table 4: Match rate between human and metric-based selection of the best θ.

talk, *talk some*, and *talk more*). Two annotators evaluated each dialogue along four dimensions using a 4-point Likert scale (1=bad, 2=fair, 3=good, 4=excellent): *correctness* of guidance, *helpfulness* of assistance, *alignment* with video content, and *naturalness* of dialogue (detailed rubrics in Appendix). The evaluation achieved a weighted inter-rater agreement of 81%, indicating strong consensus. As shown in Table 2 (top), the synthetic dialogues demonstrate consistently high quality, with average scores exceeding 3 across all dimensions. Notably, dialogue quality correlates with user interaction frequency, with more interactive dialogues scoring higher, particularly in naturalness.

Direct Comparison to Human Dialogues. To contextualize these results, we additionally evaluated human dialogues from HoloAssist and WTaG on the same samples, enabling direct comparison between synthetic and human dialogues. Table 2 (middle and bottom) shows that PROASSIST's synthetic dialogues match or outperform their human-collected counterparts across multiple dimensions. This advantage is particularly notable in the HoloAssist subset, where our generated dialogues achieve significantly higher scores in helpfulness, correctness and naturalness. Qualitative analysis reveals that human-collected dialogues often contain artifacts from Wizard-of-Oz collection setups, where untrained individuals acting as assistants may not maintain consistent professional standards. In contrast, PROASSIST dialogues are designed to emulate standardized, professional assistant interactions, resulting in more consistent and helpful guidance. These results validate the effectiveness of our data curation pipeline in producing high-quality synthetic dialogues.

6.3 Validation of Proposed Metrics

To measure whether our proposed metrics align with human judgment for model assessment, we

conducted two human evaluations.

Correlation with human preference in model ranking. We selected 50 random tasks and collected predictions from three model variants, differing in tokens per frame and access to ground-truth recipes. Annotators ranked these predictions from best to worst (allowing ties), for comparison with rankings from our pairwise F1 score and LLM overall helpfulness score. Table 3 shows that both metrics correlate positively with human judgment, with LLM scoring showing stronger alignment. We note that these correlation scores match those of previous automatic dialogue evaluation metrics (Yeh et al., 2021; Zhang et al., 2021), despite our additional challenge of measuring response timing. These results establish a baseline for developing metrics with better human correlation.

Validation of speaking threshold selection. The speaking threshold θ is a crucial hyperparameter that controls the model's balance between conservative and impulsive talking styles. To validate our use of validation F1 score for selecting θ, we compared human preferences across different thresholds with our metric-based selections. The F1 score demonstrated the highest alignment with human preferences compared to precision and recall, achieving agreement rates of 0.8 for action narration (A.N.) and 0.67 for dialogue generation (D.G.), confirming its effectiveness as a selection criterion.

6.4 Result Analysis

We analyze our experimental findings to understand the challenges of proactive task guidance and evaluate the effectiveness of our proposed techniques.

Limited gains from improved perception in dialogue generation. Table 5 shows that while increasing tokens per frame (I) substantially improves action narration performance, it provides minimal benefits for dialogue generation. This indicates that effective task guidance requires more than just better visual perception. While the model becomes better at recognizing user actions, it still needs additional capabilities–such as long-horizon progress tracking, situational reasoning, and knowl-

Model	Action Narration			Dialogue Generation						
	Precision	Recall	F1	Precision	Recall	F1	Correctness	Promptness	Efficiency	Overall
I=1	43.61	61.86	51.16	51.26	24.72	32.55	2.15	2.47	2.11	2.11
I=1 (w/ klg)	-	-	-	49.57	28.58	35.43	2.46	2.78	2.31	2.36
I=5	62.81	61.12	61.96	44.41	26.36	32.62	2.13	2.46	2.09	2.10
I=5 (w/ klg)	-	-	-	44.24	31.52	36.25	2.50	2.78	2.34	2.41
I=10	66.17	65.08	65.62	36.97	30.04	32.77	2.19	2.50	2.11	2.15
I=10 (w/ klg)	-	-	-	37.54	34.93	36.07	2.53	2.83	2.31	2.42

Table 5: Model evaluation results. Comparisons can be made across different tasks (Action Narration vs. Dialogue Generation), model variants (I=1, 5, or 10), and knowledge access setups (w/ or w/o knowledge).

Model	Action Narration			Dialogue Generation		
	Precision	Recall	F1	Precision	Recall	F1
Baseline	20.5	56.6	30.1	49.6	25.9	32.9
ρ=0.2	56.4	40.6	47.2	48.5	26.7	33.5
ρ=0.1	51.3	68.6	**58.7**	48.0	27.5	**34.4**
ρ=0.01	58.5	52.9	55.5	35.9	33.0	34.2

Table 6: Improvement from negative frame sub-sampling under different sub-sampling ratios.

Methods	Precision	Recall	F1
Drop-Middle	30.4	25.9	25.7
IPS (Ours)	49.6	25.9	**32.9**

Table 7: Comparison between inference-time context management method for long video processing.

Subset	Correctness	Promptness	Efficiency	Overall
Ego4D	2.07	2.32	2.06	2.02
HoloAssist	2.13	2.55	2.13	2.08
EgoExoLearn	1.90	2.27	1.95	1.93
Assembly101	1.94	2.24	1.98	1.93
EpicKitchens	2.07	2.26	2.04	2.02
WTaG	2.79	3.16	2.51	2.67

Table 8: Per-subset performance across domains.

edge application–to provide meaningful guidance. These results highlight the fundamental challenges in our new problem formulation and emphasize the importance of higher-level reasoning capabilities. **Benefits of task-specific knowledge.** Table 5 also shows that providing the model with ground-truth knowledge (e.g., recipes) significantly improves guidance quality across all metrics. This improvement suggests that accessing recipes enables the model to align its guidance strategy with the specific plan shown in the video. This is critical for evaluation with pre-recorded demonstrations where multiple valid solutions exist but only one is shown. Without such knowledge, the model might be penalized for suggesting equally valid alternatives. We therefore recommend the knowledge-conditioned setup as the standard configuration for our evaluation framework. Becides, these findings also highlight the importance of retrieval-augmented generation (RAG) with task-relevant knowledge to improve real-world proactive assistant systems. **Effectiveness of Negative Frame Sub-sampling (NFS).** We apply NFS with different sampling ratios ρ. As shown in Table 6, training with NFS consistently improves the model's response timing decisions, with higher F1 scores across both tasks. The optimal performance is achieved at $\rho = 0.1$, which we adopt for all subsequent experiments.

Effectiveness of Iterative Progress Summarization (IPS). Direct ablation of IPS is infeasible, as the evaluation cannot complete for videos exceeding the model's training context length. We instead compare against a modified version of StreamingLLM (Xiao et al., 2024)–a context management approach that handles memory constraints by dropping middle tokens while preserving initial task goals. Table 7 shows that IPS significantly outperforms this baseline, demonstrating its effectiveness in long-term task progress tracking.

Task familiarity impacts performance. Table 8 reveals significant performance variation across PROASSIST subsets. The model performs notably better on WTaG tasks, which contain only three unique tasks that appear in training (albeit in different environments during evaluation). In contrast, performance drops substantially for EgoExoLearn and Assembly101 tasks, due to relatively less training samples available for laboratory and assembly domains. These results highlight the need to improve generalization to new tasks and domains.

7 Conclusion

We introduce a novel framework for perceptual task guidance through streaming video dialogue generation, supported by PROASSIST–a large-scale synthetic dataset, validated evaluation metrics, and an enhanced end-to-end model. Our experiments reveal that while visual perception alone has limited impact, task knowledge and effective memory mechanisms significantly improve performance. We hope the curated data, new evaluation metrics, and our baseline models will provide much needed resources and insights, establishing a foundation

for advancing real-time AI assistance.

Limitations

Our dialogue synthesis pipeline, while carefully designed, has room for improvement in quality control. As shown in Table 2, the alignment between dialogues and video content requires enhancement. Future work could leverage more advanced LLMs, refined prompt engineering, or incorporate multimodal models to increase synthesis quality.

The dataset's reliance on pre-existing video annotations limits its scalability, as such annotations are expensive and time-consuming to obtain. Recent advances in multimodal LLMs (Achiam et al., 2023; Anthropic, 2024; Team et al., 2024) open the possibility of generating dialogues directly from raw videos, which could make data synthesis more efficient and scalable.

While our automatic evaluation metrics show promise, their validation is limited to our current experimental setup. These metrics need broader testing across diverse models, performance levels, and related tasks. Additionally, our text-only evaluation approach could be enhanced by incorporating multimodal metrics that consider video content, to establish more robust benchmarks for interactive assistant systems.

Another limitation is that we do not explicitly model utterance duration that regards precise user-assistant turn-taking simulation. However, for the core task defined in our work, determining when and how to provide proactive guidance based on streaming video context, evaluating the timing of interventions is a starting point towards more elaborate timing/duration management in the future.

Finally, while our proposed proactive assistant model is the first to tackle this challenge, its performance remains suboptimal. As shown in Table 8, even on the best-performing domain, the model falls below acceptable thresholds in LLM evaluation (overall scores below 3 out of 5). Notably, it struggles with response timing, dialogue consistency, and delivering detailed guidance that demands fine-grained perception. These limitations underscore the need for better modeling of speaking time, stronger visual-language alignment, and enhanced visual understanding in streaming dialogue generation.

Ethics Statement

All datasets used in this work are publicly available and do not contain sensitive or private information. The models used in our data synthesis pipeline and experiments are based on open-source frameworks and will be released upon publication. We acknowledge that LLM-generated utterances may exhibit hallucinations or biases, and we conduct human evaluations to understand the quality of our synthetic dialogue data.

Acknowledgments

References

Yelaman Abdullin, Diego Molla-Aliod, Bahadorreza Ofoghi, John Yearwood, and Qingyang Li. 2024. Synthetic dialogue dataset generation using llm agents. *arXiv preprint arXiv:2401.17461*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Yuwei Bao, Keunwoo Peter Yu, Yichi Zhang, Shane Storks, Itamar Bar-Yossef, Alexander De La Iglesia, Megan Su, Xiao Lin Zheng, and Joyce Chai. 2023. Can foundation models watch, talk and guide you step by step to make a cake? *arXiv preprint arXiv:2311.00738*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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*, pages 18407–18418.

Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Seokhwan Kim, Andy Rosenbaum, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. 2023. Places: Prompting language models for social conversation synthesis. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 844–868.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. 2020. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2022. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54:755–810.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ron Garland. 1991. The mid-point on a rating scale: Is it desirable. *Marketing bulletin*, 2(1):66–70.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. 2024. Ego-exo4d: Understanding skilled human activity from first-and third-person

perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. 2024. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13504–13514.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yifei Huang, Guo Chen, Jilan Xu, Mingfang Zhang, Lijin Yang, Baoqi Pei, Hongjie Zhang, Lu Dong, Yali Wang, Limin Wang, et al. 2024. Egoexolearn: A dataset for bridging asynchronous ego-and exo-centric view of procedural activities in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22072–22086.

Muhammet Ilaslan, Chenan Song, Joya Chen, Difei Gao, Weixian Lei, Qianli Xu, Joo Lim, and Mike Shou. 2023. Gazevqa: A video question answering dataset for multiview eye-gaze task-oriented collaborations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10462–10479.

Roy Jonker and Ton Volgenant. 1988. A shortest augmenting path algorithm for dense and sparse linear assignment problems. In *DGOR/NSOR: Papers of the 16th Annual Meeting of DGOR in Cooperation with NSOR/Vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR*, pages 622–622. Springer.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. 2024. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. Simmc 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Shih-Po Lee, Zijia Lu, Zekun Zhang, Minh Hoai, and Ehsan Elhamifar. 2024. Error detection in egocentric procedural task videos. In *Proceedings of the*

10

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18655–18666.

Teesid Leelasawassuk, Dima Damen, and Walterio Mayol-Cuevas. 2017. Automated capture and delivery of assistive task guidance with an eyewear computer: the glaciar system. In *Proceedings of the 8th Augmented Human International Conference*, pages 1–9.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pretraining with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023b. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Yanwei Li, Chengyao Wang, and Jiaya Jia. 2025. Llamavid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer.

Zekun Li, Wenhu Chen, Shiyang Li, Hong Wang, Jing Qian, and Xifeng Yan. 2022. Controllable dialogue simulation with in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4330–4347.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024a. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yao Lu and Walterio Mayol-Cuevas. 2019. Higs: Hand interaction guidance system. In *2019 IEEE international symposium on mixed and augmented reality adjunct (ISMAR-Adjunct)*, pages 376–381. IEEE.

Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. 2023. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. 2021. Simulated chats for building dialog systems: Learning to generate conversations from instructions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1190–1203.

Seungwhan Moon, Satwik Kottur, Paul A Crook, Ankita De, Shivani Poddar, Theodore Levin, David Whitney, Daniel Difranco, Ahmad Beirami, Eunjoon Cho, et al. 2020. Situated and interactive multimodal conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1103–1121.

Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Tushar Nagarajan, Matt Smith, Shashank Jain, Chun-Fu Yeh, Prakash Murugesan, Peyman Heidari, Yue Liu, et al. 2024. Anymal: An efficient and scalable any-modality augmented language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1314–1332.

Jennifer Ockerman and Amy Pritchett. 2000. A review and reappraisal of task guidance: Aiding workers in procedure following. *International Journal of Cognitive Ergonomics*, 4(3):191–212.

Jennifer J Ockerman and Amy R Pritchett. 1998. Preliminary investigation of wearable computers for task guidance in aircraft inspection. In *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No. 98EX215)*, pages 33–40. IEEE.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Rohith Peddi, Shivvrat Arya, Bharath Challa, Likhitha Pallapothula, Akshay Vyas, Jikai Wang, Qifan Zhang, Vasundhara Komaragiri, Eric Ragan, Nicholas Ruozzi, et al. 2023. Captaincook4d: A dataset for understanding errors in procedural activities. *arXiv preprint arXiv:2312.14556*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of*

the AAAI conference on artificial intelligence, volume 34, pages 8689–8696.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Arvin Christopher C Reyes, Neil Patrick A Del Gallego, and Jordan Aiko P Deja. 2020. Mixed reality guidance system for motherboard assembly using tangible augmented reality. In *Proceedings of the 2020 4th International Conference on Virtual and Augmented Reality Simulations*, pages 1–6.

Ayaka Sato, Keita Watanabe, and Jun Rekimoto. 2014. Mimicook: a cooking assistant system with situated guidance. In *Proceedings of the 8th international conference on tangible, embedded and embodied interaction*, pages 121–124.

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

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. 2024a. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232.

Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. 2024b. Ego4d goal-step: Toward hierarchical understanding of procedural activities. *Advances in Neural Information Processing Systems*, 36.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*.

Qingyue Wang, Liang Ding, Yanan Cao, Zhiliang Tian, Shi Wang, Dacheng Tao, and Li Guo. 2023a. Recursively summarizing enables long-term dialogue memory in large language models. *arXiv preprint arXiv:2308.15022*.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. 2023b. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20270–20281.

Xuan Wang, SK Ong, and Andrew Yeh-Ching Nee. 2016. Multi-modal augmented-reality assembly guidance based on bare-hand interface. *Advanced Engineering Informatics*, 30(3):406–421.

Zhanyu Wang, Longyue Wang, Zhen Zhao, Minghao Wu, Chenyang Lyu, Huayang Li, Deng Cai, Luping Zhou, Shuming Shi, and Zhaopeng Tu. 2024. Gpt4video: A unified multimodal large language model for lnstruction-followed understanding and safety-aware generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3907–3916.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. 2025. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pages 453–470. Springer.

Benita Wong, Joya Chen, You Wu, Stan Weixian Lei, Dongxing Mao, Difei Gao, and Mike Zheng Shou. 2022. Assistq: Affordance-centric question-driven task completion for egocentric assistant. In *European Conference on Computer Vision*, pages 485–501. Springer.

Te-Lin Wu, Satwik Kottur, Andrea Madotto, Mahmoud Azab, Pedro Rodriguez, Babak Damavandi, Nanyun Peng, and Seungwhan Moon. 2023. SIMMC-VR: A task-oriented multimodal dialog dataset with situated and immersive VR streams. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6273–6291, Toronto, Canada. Association for Computational Linguistics.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10714–10726.

Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. 2021. A comprehensive assessment of dialog evaluation metrics. In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pages 15–33.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.

Haolan Zhan, Sameen Maruf, Ingrid Zukerman, and Gholamreza Haffari. 2024. Going beyond imagination! enhancing multi-modal dialogue agents with synthetic visual descriptions. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 420–427.

Chen Zhang, Yiming Chen, Luis Fernando D'Haro, Yan Zhang, Thomas Friedrichs, Grandee Lee, and Haizhou Li. 2021. Dynaeval: Unifying turn and dialogue level evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5676–5689.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Chenchen Zhu, Fanyi Xiao, Andrés Alvarado, Yasmine Babaei, Jiabo Hu, Hichem El-Mohri, Sean Culatana, Roshan Sumbaly, and Zhicheng Yan. 2023a. Egoobjects: A large-scale egocentric dataset for fine-grained object understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20110–20120.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023b. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Dataset

A.1 Synthetic Dialogue Data Generation Details

Videos in PROASSIST are sourced from six extensively labeled datasets: Ego4D (Grauman et al., 2022) with GoalStep annotations (Song et al., 2024b), EpicKitchen (Damen et al., 2020, 2022), HoloAssist (Wang et al., 2023b), Assembly101 (Sener et al., 2022), EgoExoLearn (Huang et al., 2024), and WTaG (Bao et al., 2023). Detailed statistics and label types are summarized in Table 9. When generating timestamped video descriptions for these videos, we leverage all available labels to provide a comprehensive understanding of the video content. In cases where both coarse- and fine-grained action labels are available, they are organized into hierarchical formats for clarity. An example of this unified timestamped video description, incorporating coarse and fine-grained actions, mistake corrections, and assistant-user dialogues, is shown in List 1.

```
[11.0s-28.2s] The user grabs the GoPro.
 - [11.0s-12.2s] approach gopro
 - [13.7s] assistant: "Okay."
 - [15.1s] assistant: "You can pull the GoPro."
 - [16.9s] user: "GoPro."
 - [17.0s-22.2s] grab gopro
 - [22.2s-23.5s] flip bag
[29.9s-66.2s] The user changes the battery for
the GoPro.
 - [30.2s] assistant: "Change the Battery."
 - [34.2s-41.0s] pull battery door
 - [41.0s-42.3s] open battery door
 - [43.5s-45.0s] grab battery
 - [45.1s-46.1s] withdraw battery
 - [45.6s] assistant: "Take out the battery."
 - [47.5s-49.6s] place battery
 - [48.8s] assistant: "Now, put it down"
 - [49.7s-50.9s] lift battery
 - [50.9s-51.7s] insert battery
 - [58.8s] assistant: "Close it."
 - [59.8s-61.0s] close battery door
 - [62.2s-63.5s] push battery door
 - [67.0s] assistant: "Now change the micro SD."
[68.4s-304.3s] The user opens the GoPro.
 - [68.6s-70.8s] grab battery door
 - [70.8s-72.7s] open battery door
 - [72.7s-73.9s] press battery (ERROR: The user
 presses the wrong place.)
 - [73.9s] assistant: "No that one."
 ...
```

Listing 1: An example video description from HoloAssist.

Next we provide additional details for each step in our data curation pipeline.

Task Goal and Recipe Generation The objective is to generate a high-level task goal and recipe-style instructions that outline the key steps required to complete the task, derived from the video descriptions. For datasets where these elements are already provided in their labels (EgoExoLearn, WTaG), this step is skipped. During generation, we observe that the LLM can sometimes be distracted by irrelevant actions in the video descriptions, resulting in less accurate or unstable outputs across different sampling trials. To address this issue, we employ a two-step process. First, we generate 10 candidate recipes[4] using the prompt shown in List 2. Next, we refine these recipes into a single cohesive and integrated version by calling the LLM one more time with the prompt in List 3.

```
Here is a video description of an experienced
user working on the task - {goal_description}:
{video_descriptions}

Try to infer the **high-level** recipe from the
descriptions. Note that the steps may not
belong to the same trial, so you have to infer
the correct order of the steps based on common
sense, and re-order the steps if necessary. Do
not hallucinate details that are not mentioned
in the descriptions. Also generate a more
**informative** and **descriptive** name for
the task based on provided descriptions. The
name should be a description of the task,
instead of the name of the recipe.

Give plain and concise text with numbered key
steps in the following format:
[task name] 1. ...  2. ...
```

Listing 2: Prompt for inferring task goal and recipe from video descriptions.

```
Here are {num_repeats} {knowledge_type}s:
{recipes}

Some may be incorrect or incomplete. Please
give a single correct and complete
{knowledge_type} for the task, with numbered
key steps. Pick the title that is descriptive
for the task, instead of a {knowledge_type}
name.

Give plain, unformatted and concise text with
numbered key steps in the following format:

[task name] 1. ... 2. ...

Do not include any other information or note.
```

Listing 3: Prompt for task goal and recipe refinement.

Video Pre-Filtering The next step is to filter out videos that are unsuitable for proactive assistant-user dialogue modeling. First, we exclude videos

---

[4]We use vLLM(Kwon et al., 2023) for efficient parallel sampling.

Dataset	Domain	#Tasks	#Videos	Total Duration	Avg Duration	Labels
Ego4D-Goalstep (Grauman et al., 2022; Song et al., 2024b))	Cooking	86	851	368h	26m	C + F
EpicKitchen (Damen et al., 2020, 2022)	Cooking	-	700	100h	9m	F
HoloAssist (Wang et al., 2023b)	Object Manipulation; Assembly	20	2221	166h	5m	C + F + M + D
EgoExoLearn (Huang et al., 2024)	Cooking; Laboratory Tasks	8	432	96h	13m	C + F + R
Assembly101 (Sener et al., 2022)	Assembly	101*	356	42h	7m	C + F + M
WTaG (Bao et al., 2023)	Cooking	3	56	10h	11m	C + M + D + R

Table 9: Summary of egocentric video datasets used in PROASSIST. The statistics presented are as originally reported in the corresponding papers before filtering. The number of tasks indicates the types of tasks in each dataset, except for Assembly101, where it represents the unique number of toys. Label abbreviations are as follows: C (coarse action labels), F (fine-grained action labels), M (mistake and correction labels), D (human-collected assistant-user dialogues), and R (ground-truth recipes).

with low label coverage, as their descriptions may lack sufficient detail to provide a clear understanding of the content. Then, using the video descriptions, task goals, and recipes generated in the previous step, along with the domain and recipe type derived from the dataset metadata, we prompt the LLM to classify each video into one of three categories:

- 0: The task does not belong to the target domain.

- 1: The camera wearer performs the target task following the recipe.

- 2: The camera wearer performs other tasks simultaneously while working on the target task.

We keep only the videos classified as category 1 for subsequent steps. To minimize noise, this classification process is repeated 10 times for each task, and the majority label is used as the final classification.

```
Here is a video description of an user working
on the task - {goal_description}:
{video_descriptions}

Reference {knowledge_type}:
{knowledge}

Is this a {knowledge_type}? If so, was the user
likely to:
1. perform the task roughly following the
{knowledge_type} (**no** need to be strict), OR
2. perform other tasks (or another trial of the
same task) simultaneously in a multi-tasking
manner?

Answer with your analysis, and end your
response with "Final answer: 1, 2 or 0" (0
denotes that the activity is not related to
{domain}).
```

Listing 4: Prompt for video pre-filtering.

Multi-Round Dialog Generation   To simulate realistic dialogues aligned with video content, we design a detailed instruction prompt, as shown in List 5. We incorporate dataset-specific instructions to account for variations across data sources to improve generation quality (List 6). To simulate diverse user behaviors, we define three type of user profiles that provide high-level guidelines for user interaction:

- `no_talk`: The user follows the assistant's instructions without speaking.

- `talk_some`: The user occasionally asks questions or seeks confirmation about instructions, accounting for approximately 20% of the steps.

- `talk_more`: The user is talkative, asking both task-related and unrelated questions, accounting for approximately 40% of the steps.

Since we can sample different user behaviors for the same video, our synthetic dataset can be easily expanded by generating multiple variations. Specifically, we create 10 dialogues per video, distributed across user types in a 2:4:4 ratio. In practice, we observe that very long videos can result in excessively lengthy prompts, which may lead to poor alignment between the video description and the generated dialogue due to performance degradation of LLMs when processing long contexts (Liu et al., 2024b). To address this, we propose an iterative approach to generate dialogues within a limited time window chunk by chunk. For each chunk, we provide only the video description corresponding to that time window and up to the 10 most recent dialogue turns to ensure contextual consistency. This modification significantly improves alignment between the video description and the generated dialogue while stabilizing memory consumption due to the reduced prompt length. After generation, we conduct an additional refinement

step to enhance dialogue quality. Specifically, we prompt the LLM to merge dialogue turns that occur close in time, improve naturalness and fluency by incorporating more coreference and pronouns, make assistant responses more concise, and avoid unfriendly behaviors. Refer to List 7 for details.

```
Here is a video description of an user working
on the task - {goal_description}:
{video_descriptions}

Your goal is to simulate a conversation between
the user and an assistant, where the user's
actions are performed following the assistant's
instructions. The user will first mention the
overall goal of the task. The assistant informs
the user about the next step at proper time.
Importantly, the assistant is proactive and
always provides the next step even before the
user asks for it. Before the task starts, the
assistant may also give a brief introduction
about the task. {additional_requirement}

Requirements for the assistant:
- Time is crucial! Try to generate the dialog
that strictly aligns with the video timeline.
- Try to cover all the essential steps in the
task. If the user asks a question at the time
the assistant should give the next step, the
assistant turn should include both the response
to the question and instruction about the next
step.
- Be helpful and friendly. If the user asks
something that has been explained before, the
assistant should still provide the information
with patience.
- Try to be encouraging when the user makes
progress, but do not overdo it.
- Be concise! The dialog is verbal, so avoid
long sentences.
- Do not say "can you do it for me" to the user.


Requirements for the user:
{user_requirement}


Generation format:
[time] User: ...
[time] Assistant: ...
[time] Assistant: ...
[time] User: ...
[time] Assistant: ...

Note that the minimal interval between each
turn is 1 second, which means the user will
wait for at least 1 second after an assistant's
turn, and two consecutive assistant's turns
should have at least 1 second interval. Combine
close turns into a single turn if necessary.
One exception is that the assistant must
respond **immediately** when the user says
something (i.e. give a response right after an
user's turn at the same time).

{dialog_history}

In this round, please **only** generate the
dialog for the video from time
```

```
[{start_time:.1f}s] to [{end_time:.1f}s]!
```

Listing 5: Prompt for dialogue simulation.

```
HoloAssist: Note that the video description
contains both the user's actions and the
user-assistant dialog. Anchor the simulated
dialog to the existing dialog, and try to
rephrase the utterances to make them more
coherent and human-like. You may add a few more
turns around the **essential steps** of the
task, which are the underlying intentions of
the action instead of the actions themselves.
Add a few turns to make the dialog more fluent
and helpful, but avoid being overwhelming.

EgoExoLearn: The simulated dialog should be
centered around the **key steps** of the task,
not every single action of the user. Try to
make the dialog more coherent and helpful as
what a human assistant will say.

Epickitchens: The simulated dialog should be
centered around the **key steps** of the task,
not every single action of the user. Note that
the user may make mistake or perform suboptimal
actions, the assistant should not give
instructions on those actions, but smartly
select right time to give guidance. Try to make
the dialog more coherent and helpful as what a
human assistant will say.

WTaG: Note that the video description contains
both the step description and the
user-assistant dialog. Anchor the simulated
dialog to the existing dialog, and try to
rephrase the utterances to make them more
coherent and human-like. Add more details such
as assistant feedback or user question during
long steps if necessary. Remember to generate
the response to user's question even if there
isn't one in the original dialog from the video
description.

Assembly101: The mistakes made by the user are
marked by (mistake: <mistake type>). If a
mistake happens, we want to simulate the dialog
in the way that the assistant helps the user
correct the mistake. To be more specific, the
assistant SHOULD NOT give instructions if an
action is 'wrong order', 'previous one is
mistake' or 'shouldn't have happened'. Instead,
the assistant should give instruction of the
CORRECT next step (i.e. scan the future actions
and select the nearest correct action).
Afterwards, at the start of actions marked as
'correction', the assistant should mention the
previous mistake and give insruction on how to
correct it based on the corrective action. For
'wrong position' mistakes, the assistant can
give the instruction of that action, but need
to point out the mistake at the start time of
corrective action for that mistake.
```

Listing 6: Prompt specific for each dataset as additional requirements.

```
Here is a conversation between a user and an
assistant:
```

3

```
{dialog_history}

For each assistant message, add labels
regarding the assistant's initiativity and
intention:

Initiativity:
- initiative: The assistant says something
proactively without the user asking for it.
- responsive: The assistant responds to the
user's question or comment.

Intention:
- instruction: The assistant gives an
instruction to the user.
- correction: The assistant corrects a mistake
made by the user, either proactively or
responsively. Suggestions for alternative
actions can also be included.
- info_sharing: The assistant shares some
information with the user, such as explaining
something or giving a tip.
- feedback: The assistant gives feedback to the
user, such as "good job" or "tips for
improvement".
- other: Other intentions that do not fall into
the above categories.

Intention can be multiple, e.g., "instruction,
info_sharing".

Generation format:
[time] User: ...
[time] Assistant: ... [initiativity|intentions]
[time] Assistant: ... [initiativity|intentions]
[time] User: ...
[time] Assistant: ... [initiativity|intentions]

When generating the dialog, you should also
refine the dialogue following these guidelines:
1. Merge turns that are close in time (less
than 1 second apart) into a single turn, when
the content is similar or related.
2. Use more coreference and pronouns to make
the dialog more coherent and human-like.
3. Decide the length of assistant messages
smartly. Make them more clear and helpful when
necessary, but keep them concise and to the
point in general.
4. Avoid repeating the same talking patterns or
phrases. For example, do not say "make sure
..." for every instruction.
5. Rephrase impolite or inappropriate language,
such as "as I have mentioned this earlier ...",
to be more friendly and helpful. But keep
concise and to the point.
6. Remove anything other than the dialog
itself, such as the user's actions or
explanations of how the dialog is generated.
Do not just copy paste the original dialog!
```

Listing 7: Prompt for dialogue refinement and intent labeling for assistant turns.

Dialogue Annotation   To facilitate the analysis of our generated dialogues, we use LLM to annotate the initiativity (responsive or initiative) and intention type (instruction, correction, information sharing, feedback, and other) of each assistant turn. We find such annotation can be effectively generated within the dialogue refinement step, using a single LLM call with the prompt in List 7. Additionally, we generate progress summaries at each assistant turn to support the iterative progress summarization approach (§5.3. These summaries include details such as the elapsed time, the task goal mentioned by the user, completed steps as progress, topics discussed by the user, and the current state or step of the task (List 8).

```
Here is a conversation between a user and an
assistant:
{dialog_history}

Summarize the task goal and progress so far,
including:
1. The task goal mentioned by the user.
2. What has been done.
3. Other topics mentioned by the user in the
conversation, if any.
4. The current state/step of the task.
Be faithful and try to include all the relevant
information.

Give your response in plain text of a single
line in the following format:
SUMMARY: <progress summary>
```

Listing 8: Prompt for progress summary generation

Automatic Quality Evaluation   To evaluate the quality of the generated dialogues, we assess their alignment and step coverage with the corresponding video descriptions. We first extract all time steps from the video descriptions, denoted as T_v, and all time steps from the generated dialogues, denoted as T_d. For each time step in T_d, we identify its closest time step in T_v and compute the average time difference across all pairs, normalized by the number of dialogue turns. Similarly, for each time step in T_v, we find its closest match in T_d and calculate the average time difference, normalized by the number of video description steps. These values approximate the precision of dialogue turns relative to the video (p) and the recall of task steps in the video descriptions (r). Additionally, to ensure the assistant remains responsive, we count the number of user turns without an immediate assistant response as a penalty term nr. The final quality score is computed as $score = 10 - p - r - nr$, the higher the better.

Post-Filtering and Data Splitting   We derive our training set from the training splits provided by each original dataset, while our validation and testing sets are based on the respective validation splits.

For the training set, we filter out dialogues with a score below 3. For the validation sets, we retain only the highest-scoring dialogue for each user type. If any dialogue for a video scores below 5, the video is removed. From the remaining videos, where each has three dialogues, we evenly split them into validation and test sets. This process removes approximately 25% of the dialogues and 41 hours of video.

A.2 Implementation Details

We utilize LLaMA-3.1-70B-Instruct (Dubey et al., 2024) as the LLM for all the steps described above. The model is hosted locally using vLLM (Kwon et al., 2023), running a FP8-quantized version[5] on four H100 GPUs. Although we use a specific LLM for data generation, our pipeline is model-agnostic and can be readily adapted to more advanced models with minimal prompt modifications. We hope our open-sourced prompt designs will support future efforts in curating higher-quality datasets with more capable LLMs.

A.3 Data Statistics and Distributions

Figure 5 provides a comprehensive overview of the PROASSIST data statistics. Specifically, Figure 5a presents the distribution of task domains by video durations, showing that the dataset predominantly contains cooking tasks (58.3%), followed by object manipulation (25.4%), assembly (12.0%), and laboratory tasks (4.3%). Figure 5b illustrates the dialogue length distribution, measured by the number of turns per dialogue, highlighting a significant variability with some dialogues exceeding 200 turns. Figure 5c shows the task duration distribution where the majority of tasks last under 20 minutes, while some extend up to an hour. Figure 5d visualizes the length distribution of user and assistant utterances, as well as the generated progress summaries. Assistant utterances are generally longer (mean = 16.1 words) compared to user utterances (mean = 6.5 words). For the generated summaries, the average length is 91.3 words, with the largest summary less than 200 words, showing that the information can be successfully compressed into such summarizes without growing linearly with the dialogue. Figure 5e highlights assistant initiativity and intention distributions across three user types. As users speak more frequently, the assistant responds with more reactive utterances as expected.

Regarding assistant intentions, the majority of utterances provide instructions (over 60%), with information sharing and feedback also being common. Mistake corrections occur less frequently, primarily because most of the videos used are error-free, underscoring the need to collect more task execution videos that include mistakes.

A.4 Examples

See Figure 6-11 or example dialogues from PROASSIST. Due to space constraints, only the first 12 dialogue turns are displayed for each task.

B Additional Details of Evaluation Metrics

B.1 Pairwise Evaluation

We utilize the all-mpnet-base-v2[6] model from the Sentence-Transformers library (Reimers and Gurevych, 2020) to compute text similarity. A similarity threshold of 0.5 is applied to determine correct matches. For temporal cost calculation, the cost-increasing rate p is set to 1.5. The cutoff range R is determined based on the average speaking interval of each dataset: 2.5 seconds for action narration and 1.5–6.0 seconds for dialogue generation. In particular for dialogue generation, we set $L = R/2$ to penalize delayed predictions more heavily, because it is important for the model to provide instruction of the next step before the user begins performing it in the video.

B.2 LLM-based Evaluation

The detailed description of each metric and rubric of 5-scale Likert score is described in List 9, which is also the prompt given to the LLM judge. To reduce the randomness of scoring, we repeat each evaluation for 3 times, and use the average score as the final score for each metric. We use LLaMA-3.1-70B-Instruct as the evaluator in our experiments.

```
You are an expert in evaluating the quality of
user-assistant dialogues. Your task is to
evaluate dialog responses generated by an
assistant model that helps users with their
tasks. You should evaluate the dialogs by
comparing them to reference gold-standard
dialogues from professional assistants.

Requirement:
1. Read dialogues carefully and compare them
line by line. Keep you analysis concise and to
the point.

2. Evaluate the following aspects:
```

(a) Task Domain Distribution.

(b) Dialogue length distribution.

(c) Task duration distribution.

(d) Length distribution for assistant, user utterances and our generated progress summary.

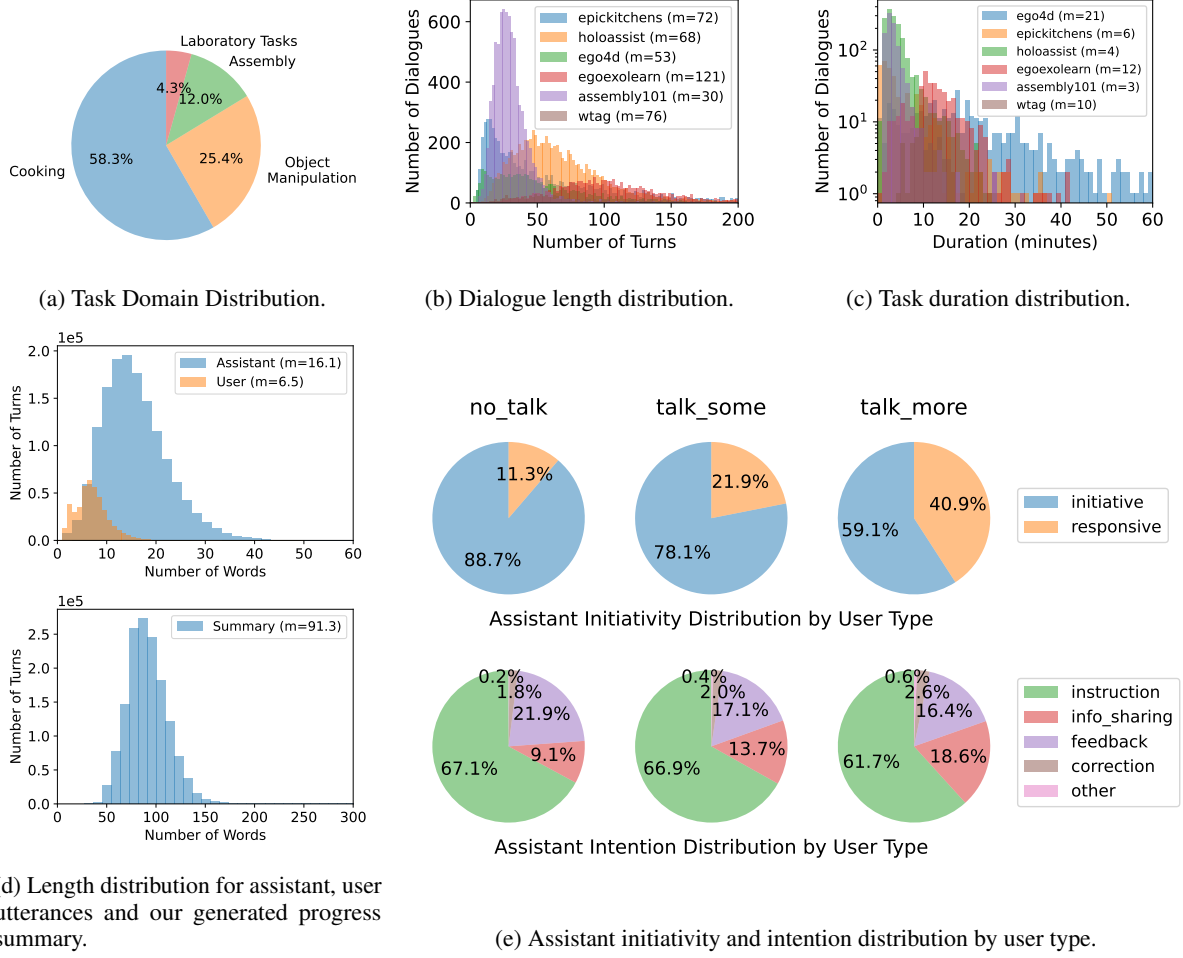(e) Assistant initiativity and intention distribution by user type.

Figure 5: PROASSIST dataset statistics overview including task domain distribution, dialogue structure, duration variability, and assistant interaction patterns. Mean values for each distribution are provided in the legend.

```
- Correctness: does each generated
instruction/feedback make sense (correct or
relevant) or not, based on the context and the
gold-standard reference?
- Promptness: does the assistant provide
guidance at the right time, or does it talk too
early or too late?
- Efficiency: does the assistant provide the
necessary information in a concise and
efficient manner, without too much repetition
or redundancy information?
- Overall: the overall helpfulness and quality
of the assistant's responses.

3. For each aspect, give a score from 1 to 5
based on the following criteria:
- 1=very poor: most of utterances are
incorrect, irrelevant, mistimed, inefficient etc
- 2=poor: bad utterances that are incorrect,
irrelevant, mistimed are more than good ones
- 3=average: the number of good and bad
utterances are roughly the same
- 4=good: more good utterances than bad ones
- 5=excellent: most of utterances are correct,
relevant, timely, efficient etc
```

Listing 9: Prompt for LLM-based end-to-end evaluation.

C Model Implementation Details

C.1 VideoLLM-Online Model Implementation

We use LLaMA-3.1-8B-Instruct (Dubey et al., 2024) as the base LLM and the pretrained SigLIP-SO400M-14-384[7] model (Zhai et al., 2023) as the frame encoder. To extract frame features, we use the embeddings from the second last layer of the [CLS] token, and $N \times N$ patch features obtained through average pooling of the corresponding patch embeddings. In our experiments, we test with three model variants with $N = 0, 1, 2$, resulting in $I = 1, 5, 10$ tokens per frame, respectively. We use a two-layer MLP as the projector to project the visual features into the LLM's embedding space following (Chen et al., 2024). We remove the separator token between frames because we find it does not help with the performance.

---
[7] https://huggingface.co/google/siglip-so400m-patch14-384

Dataset	×	I=1			I=5			I=10		
		Ori. Size	Final Size	Proportion	Ori. Size	Final Size	Proportion	Ori. Size	Final Size	Proportion
Dialogue				**47.2%**			**45.9%**			**46.8%**
PROASSIST-Ego4D	2	4795	9590	6.2%	12350	24700	9.1%	20718	41436	9.9%
PROASSIST-HoloAssist	2	7645	15290	9.9%	10957	21914	8.1%	16116	32232	7.7%
PROASSIST-EgoExoLearn	2	5659	11318	7.3%	11414	22828	8.4%	18727	37454	8.9%
PROASSIST-EpicKitchens	2	8051	16102	10.4%	12901	25802	9.5%	19320	38640	9.2%
PROASSIST-WTaG	6	929	5574	3.6%	2051	12306	4.5%	3537	21222	5.1%
PROASSIST-Assembly101	2	7503	15006	9.7%	8514	17028	6.3%	12738	25476	6.1%
Summarization				**14.4%**			**13.2%**			**13.2%**
PROASSIST-Summary	2	11103	22206	14.4%	17925	35850	13.2%	27612	55224	13.2%
Action Narration				**18.0%**			**23.0%**			**24.2%**
Ego4D-Narration	1	27719	27719	18.0%	62350	62350	23.0%	101672	101672	24.2%
Auxiliary				**20.4%**			**17.9%**			**15.8%**
Something-Something-V2	10	1320	13200	8.6%	2639	26390	9.7%	3959	39590	9.4%
LLaVA-Pretrain	2	5598	11196	7.3%	6714	13428	4.9%	7840	15680	3.7%
EgoObjects	20	355	7100	4.6%	434	8680	3.2%	552	11040	2.6%
Total			154k			271k			420k	

Table 10: Detailed training data statistics for different model variants under the maximum sequence length of $L = 4096$. We report the upsampling ratio ($\times$), the original dataset size after splitting and packing under different I, the final dataset size after upsampling, and the proportion of each data source in the final mixture. The final data size grows with the number of tokens used to encode each frame/image.

C.2 Training

Training Datasets Our models are trained on a mixture of datasets:

- **PROASSIST**: We use two variants of the dataset, both with and without the task recipe provided as additional knowledge, to enable the model learning and adapting to both setups simultaneously.

- **PROASSIST-Summary**: To enhance summarization capabilities, we construct a video summarization dataset from PROASSIST. In this dataset, assistant dialogues are removed, leaving only user dialogues and system prompts. The learning objective is to generate the progress summary at the end. This setup requires the model to generate summaries directly from video and user inputs, avoiding reliance on ground-truth assistant dialogue context as a shortcut.

- **Online Action Narration**: As described in §6.1, this task focuses on real-time narration of the camera wearer's actions from live video streams. We reformat the action narration labels from Ego4D (Grauman et al., 2022) into the PROASSIST dialogue style (e.g., "*Assistant: C opens the fridge*"), with "*C*" denoting the camera wearer. To prevent data contamination, we exclude all videos from the validation and test sets of any Ego4D challenges.

- **Auxiliary Vision-Language Datasets**: We also incorporate several additional datasets to improve vision-language alignment: image captioning data from LLaVA (Liu et al., 2024a), action recognition from Something-Something-V2(Goyal et al., 2017), and egocentric object detection data from EgoObjects.(Zhu et al., 2023a). We repurposed the labels into dialogue format and train the model to generate them given a specific system prompt for each task.

Data Preprocessing We extract video frames at a rate of 2 frames per second (FPS) and align the dialogue timestamps with the corresponding frames. The streaming video-dialogue data is preprocessed into sequences of interleaved image and text tokens, as illustrated in Figure 3. We use a maximum sequence length of $L = 4096$ tokens in our experiments. Sequences are constrained to fit within this length, and we aim to make their lengths as close to L as possible to minimize padding and improve computational efficiency. To achieve this, for long videos in PROASSIST, we split them and inject progress summarization prompts as described in §5.3. For auxiliary vision-language datasets, multiple samples are packed into a single sequence of interleaved image-text format, such as <IMAGES><Text><IMAGES><Text>.... Here, images can consist of single or multiple frames, and text can represent image captions, action descriptions, or object descriptions. This packing strategy

	Ego4D-Narration	Ego4D	HoloAssist	EgoExoLearn	Assembly101	EpicKitchens	WTaG
I=1	0.3	0.3	0.3	0.3	0.3	0.2	0.3
I=1 (w/ klg)	0.3	0.3	0.3	0.3	0.3	0.2	0.4
I=5	0.3	0.3	0.3	0.4	0.3	0.3	0.4
I=5 (w/ klg)	0.3	0.3	0.3	0.4	0.3	0.2	0.5
I=10	0.3	0.4	0.4	0.4	0.3	0.2	0.4
I=10 (w/ klg)	0.3	0.3	0.4	0.4	0.3	0.3	0.4

Table 11: Selected speaking threshold θ for each model on each subset. We evaluate a series of θ values (0.1, 0.2, ...) for each setup and select the optimal threshold based on the appearance of a local maximum in F1 score as θ increases.

significantly reduces the number of samples compared to the original size. Since each frame is encoded into a varying number of tokens (1, 5, or 10), the same number of images can produce different token counts. Consequently, we apply the splitting and packing strategy separately for each setup, where larger I values result in more samples in the final training set. To balance the scale differences among data sources, smaller datasets are upsampled to achieve a more balanced mixture ratio. The final training data statistics are summarized in Table 10. Each data sample comprises a sequence of interleaved image-text tokens, potentially including hundreds of images, which presents significant challenges for data loading during training. To address this, we pre-extract image features using our image encoder and store them on disk. During training, we load these pre-extracted features directly rather than performing feature extraction on-the-fly, resulting in a $6\times$ speedup.

Training Strategy We adopt a single-stage training approach following recent practices (Karamcheti et al., 2024; Tong et al., 2024; Chen et al., 2024). During training, we freeze the image encoder, tune all parameters in the projector layers, and perform parameter-efficient tuning of the LLM using LoRA (Hu et al., 2022) with $r = 128$ and $\alpha = 256$. The AdamW optimizer (Loshchilov and Hutter, 2017) is used with a learning rate of $2e-4$ and 100 warmup steps. We employ a global batch size of 256, 384, and 512 for $I = 1$, 5, and 10, respectively. All models are trained for 4 epochs on the mixed dataset described above. We use $8\times$H100 GPUs for training.

C.3 Inference

As described in §6.1, a speaking threshold θ is used to decide whether to speak at each time step, where the model only decides to remain silent if the probability of predicting the [EOS] token exceeds

θ. We observe that the quality of model predictions is highly sensitive to the choice of θ. In our experiments, we perform inference multiple times with a series of thresholds on the validation split of each subset to determine the optimal θ for each model. Given our observation that model performance, in terms of F1 score, follows an inverse U-shaped curve as θ increases (Figure 4), we select the θ that yields the best local maximum of the F1 score. Table 11 summarizes the selected θ values for each subset. While this selection strategy aligns reasonably well with human judgment (as shown in Table 4), the chosen threshold is optimal for average performance across a set of videos rather than for individual tasks. Additionally, in real-world scenarios, a support set for hyperparameter tuning may not always be available. We leave the development of a better θ selection strategy for future work.

D Human Evaluation

IRB Approval The human evaluation process was reviewed and approved by the Institutional Review Board (IRB) of our institution before the experiment started. The participants have all reviewed and signed the consent forms which can be provided upon request.

Synthetic Data Quality Evaluation The set of questions and rubrics presented to human evaluators is detailed in List 10. The evaluation consists of six questions: four assess the quality of the synthetic dialogues, and two evaluate the accuracy of the generated task goals and recipes. A 4-point Likert scale is used to eliminate a neutral option, encouraging evaluators to express definitive preferences and provide more decisive judgments (Garland, 1991). The evaluation of human-collected dialogues follows the same interface, with evaluators blinded to the source of the dialogue. Each

dialogue is independently evaluated by two separate evaluators.

```
Q1 (Dialogue Correctness): Are the assistant's
instructions or answers factually correct?
1: Incorrect or misleading.
2: Mostly correct but with key errors.
3: Correct with minor issues.
4: Fully correct and precise.

Q2 (Dialogue Helpfulness): How helpful and easy
to follow is the assistant's instruction?
1: Confusing and unhelpful.
2: Unnecessary and adds little value.
3: Helpful but hard to follow.
4: Helpful and easy to follow.

Q3 (Dialogue Alignment): Does the dialogue stay
aligned with the video content in real-time?
1: Misaligned with the video content.
2: Mostly aligned but with noticeable missteps.
3: Aligned with minor timing issues.
4: Perfectly aligned with the video.

Q4 (Dialogue Naturalness): Does the dialogue
sound natural and conversational?
1: Stilted or unnatural.
2: Somewhat natural but awkward in places.
3: Mostly natural with minor awkwardness.
4: Flows naturally and is fully conversational.

Q5 (Task Goal Accuracy): Does the task goal
accurately reflect the user's intended task?
1: Inaccurate or completely misses the intended
task.
2: Mostly relevant but has key errors.
3: Accurate with minor issues.
4: Fully accurate and precise.

Q6 (Recipe Accuracy): Does the recipe
accurately describe the steps needed to
accomplish the task?
1: Inaccurate or misleading.
2: Mostly accurate but with notable errors.
3: Accurate with minor errors.
4: Fully accurate and clear.
```

Listing 10: Evaluation questions and rubrics for human evaluation on the synthetic data quality of PROASSIST.

Metric Alignment with Human Preference For the correlation experiment in Table 3, we collect generated dialogues from three models: $I = 1$, $I = 10$, and $I = 10$ (w/ klg), on the same set of tasks randomly sampled from our dataset. These generated dialogues are presented side-by-side with the ground-truth dialogues to human evaluators, who are asked to rank the generated dialogues from best to worst, allowing ties. For each task, we derive three pairwise comparison results from the human rankings. Similarly, pairwise comparison results are derived from the rankings obtained using either the F1 score or the LLM-assigned Overall score. Finally, we compute the correlations between these metrics based on the pairwise comparison results. We use a similar evaluation interface for the best-picking experiment in Table 4, with the only difference being that evaluators are asked to select the single best model instead of ranking all models. The match rate is calculated as the proportion of cases where the best model selected using our proposed metrics aligns with the human selection.

Figure 6: Example of a cooking task in PROASSIST.

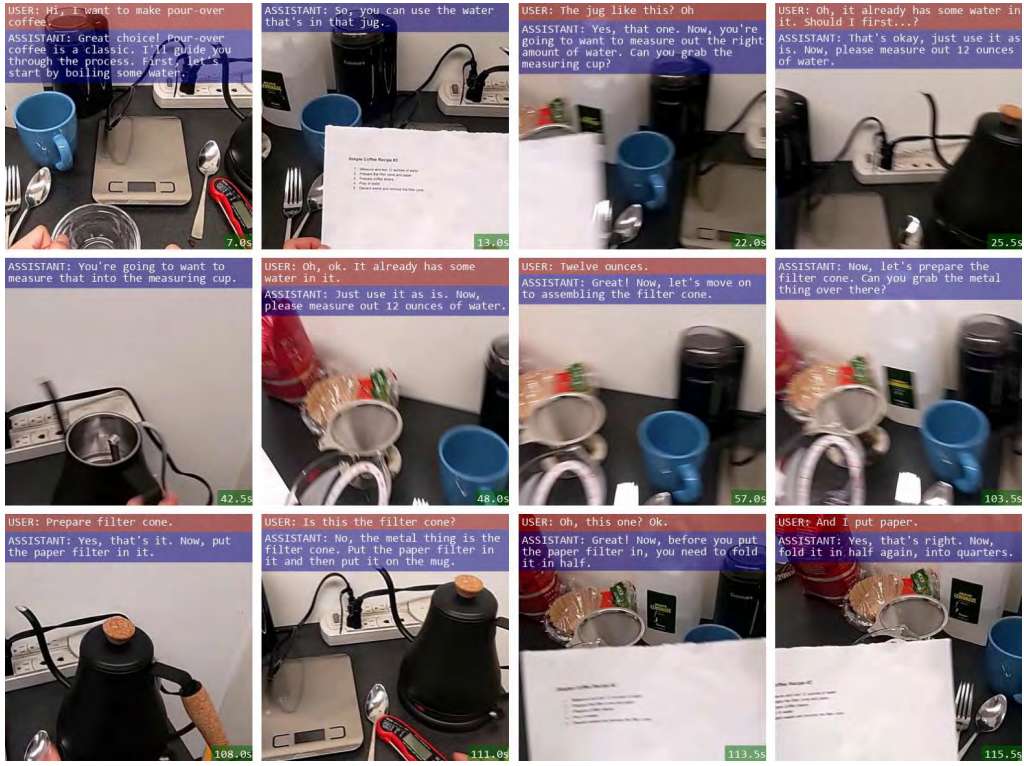Figure 7: Example of a cooking task in PROASSIST.

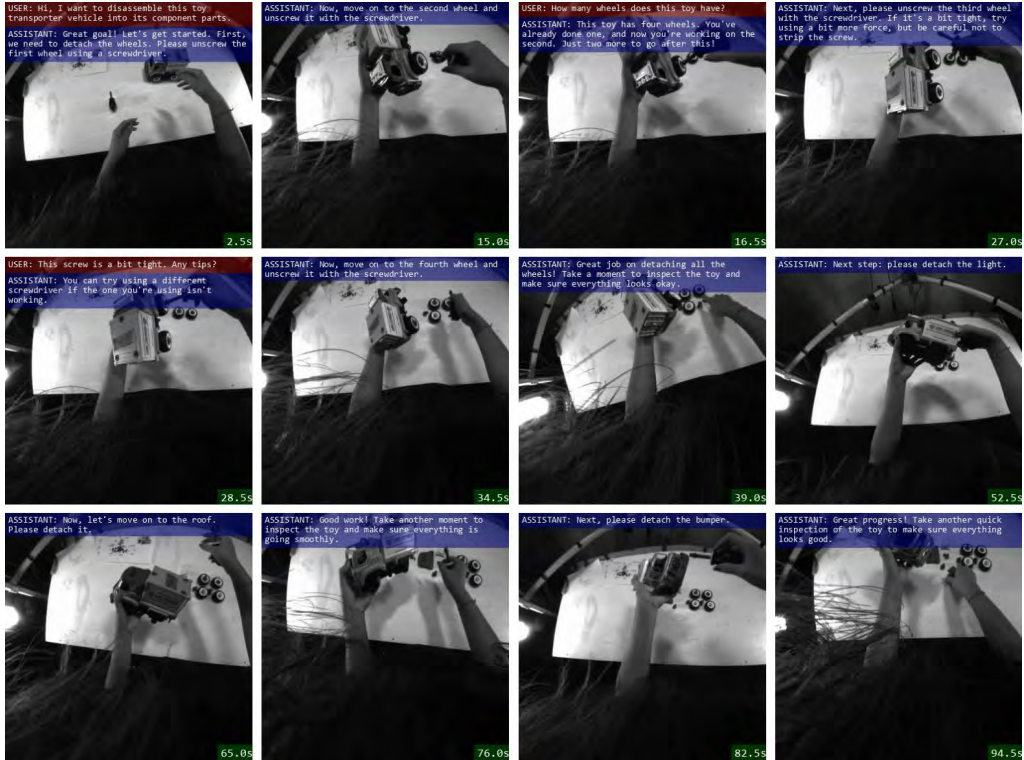Figure 8: Example of a cooking task in PROASSIST.

Figure 9: Example of a assembly task in PROASSIST.

Figure 10: Example of a laboratory task in PROASSIST.

Figure 11: Example of a cooking task in PROASSIST.