EgoVoice: Proactive Spoken Assistance from Egocentric Multimodal Streams

Heeseung Kim
Department of AI, University of Seoul
gmltmd789@uos.ac.kr

Abstract

Wearable augmented reality (AR) assistants are moving toward continuous real-world interaction, where they perceive the user's activity through first-person video and audio and provide timely spoken guidance without being explicitly asked. While proactive video assistants, spoken dialog systems, and egocentric task understanding have each advanced rapidly, existing systems do not address the joint problem of deciding when to speak and what to say from continuous first-person streams. We introduce EgoVoice, a framework for training and evaluating proactive egocentric spoken assistants. From HoloAssist video recordings of real human instructors, we construct clean audio streams through source separation and speech resynthesis, and convert each video session into a format where the model must decide at each moment whether to remain silent or provide spoken guidance. We fine-tune an omni-modal LLM with our data, and further improve its proactive intervention behavior with direct preference optimization. Experiments across closed and open-source models show that existing systems rarely produce well-timed, meaningful proactive interventions, while EgoVoice yields clear improvements in intervention timing, content relevance, and human preference over the zero-shot backbone.[1]

1 Introduction

As wearable devices such as smart glasses become more capable, augmented reality (AR) assistants are emerging as a practical interface for hands-free interaction. Meta Ray-Ban smart glasses combine voice interaction with first-person visual context, and research prototypes like Google Project Astra explore richer multimodal reasoning from egocentric camera feeds. The long-term vision goes further: an always-on assistant that perceives the user's ongoing activity through first-person (egocentric) video and audio, and provides spoken guidance at the right moment without being explicitly asked.

Despite growing interest in this direction, most voice interaction systems are designed and evaluated as reactive assistants. The user explicitly asks a question or gives a command, and the model generates a spoken or textual response (Long et al., 2026; Xie and Wu, 2024a; Zeng et al., 2024). In practice, however, real-world assistance often requires help before the user asks for it. During procedural tasks such as assembling furniture or following a recipe, users may be occupied, uncertain, or unaware that they need guidance. A user may not know that a mistake is about to occur or may be unable to pause and ask for help. This motivates *proactive egocentric spoken assistance*, where an assistant observes a first-person video and audio stream and decides whether to speak, when to speak, and what to say.

Building such an assistant requires three capabilities that have so far been studied separately: multimodal perception with speech output, simultaneous listening and speaking, and proactive intervention from streaming egocentric video. Recent multimodal models can process video, audio, and text while generating spoken responses in real-time (Fu et al., 2026; OpenAI, 2024; Xu et al., 2025; Cui et al., 2026). Full-duplex spoken dialog systems allow models to listen and speak simultaneously without explicit turn boundaries (Défossez et al., 2024; Roy et al., 2026; Wang et al., 2025a; Yu et al., 2026). Proactive video assistants can decide when and what guidance to offer from streaming egocentric video, though they produce text rather than speech (Zhang et al., 2025c; Wang et al., 2026; Chen et al., 2024; Qian et al., 2025; Zhang et al., 2026). However, to the best of our knowledge, no existing work combines these three capabilities to train a single model for proactive spoken guidance from continuous egocentric streams.

To address this gap, we introduce EgoVoice, a framework for training and evaluating proactive

[1]Project page: https://egocentricvoice.github.io/

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 29112–29136
October 24-29, 2026 ©2026 Association for Computational Linguistics

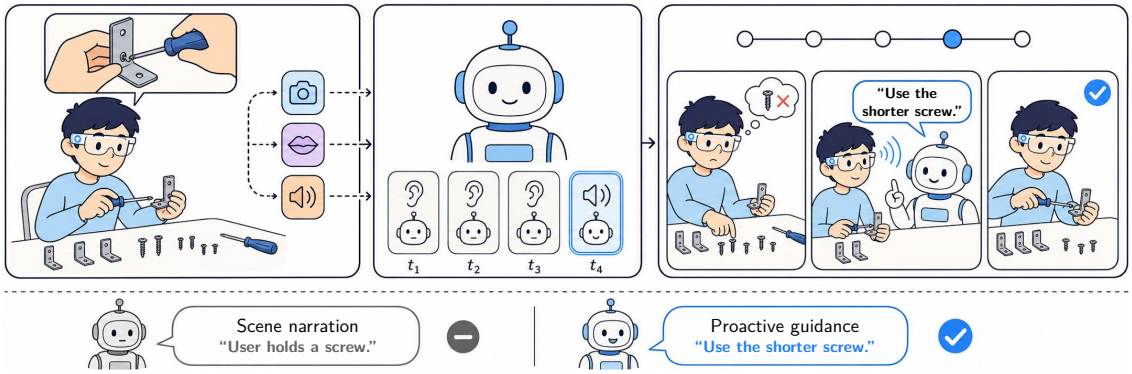


Figure 1: Overview of proactive egocentric spoken assistance. An AR assistant receives first-person video and audio (left), decides whether to remain silent or intervene (center), and provides spoken guidance (right).

egocentric spoken assistants. We start from HoloAssist (Wang et al., 2023), a corpus of egocentric task assistance with real human instructors, and process its recordings into clean, time-aligned streaming supervision through source separation and speech resynthesis. We then perform supervised fine-tuning (SFT) on MiniCPM-o 4.5 (Cui et al., 2026), an open-source full-duplex multimodal LLM, to learn when and what to say from continuous egocentric video and audio streams, and further refine intervention timing and frequency through direct preference optimization (DPO) (Rafailov et al., 2023). We evaluate models along three axes: whether they intervene at the right time, whether the content is relevant, and whether the overall assistance is helpful as judged by a large language model (LLM) and by human evaluators.

Through experiments across closed-source, open-source, and our own trained models, we show that existing multimodal systems rarely produce well-timed proactive interventions, while EgoVoice yields improvements over all baselines on timing and content relevance, with complementary strengths between the SFT and DPO variants. Our contributions are as follows:

- We construct a dataset and benchmark for proactive egocentric spoken assistance from HoloAssist recordings, producing clean training data with time-aligned intervention labels and spoken guidance targets.
- We train EgoVoice-SFT and EgoVoice-DPO, and show that the two stages are complementary: SFT retains an edge in content relevance while DPO refines intervention timing.
- We establish an evaluation protocol covering intervention timing, content relevance, and overall assistance quality for spoken assistant scenarios, and provide a comprehensive comparison across baselines and our models.

2 Related Work

Spoken Dialog and Omni-Modal Models. Early voice interaction systems relied on cascaded pipelines of automatic speech recognition (ASR), language modeling, and text-to-speech (TTS) (Lin et al., 2024). Subsequent work showed that LLMs can directly generate discrete speech tokens (Long et al., 2026; Xie and Wu, 2024a; Zeng et al., 2024; Zhang et al., 2023; Kim et al., 2024; Fang et al., 2025b). This line has rapidly scaled to omni-modal models that accept interleaved visual data, audio, and text while producing both text and speech in a streaming manner (Fu et al., 2026; OpenAI, 2024; Xu et al., 2025; Cui et al., 2026; Xie and Wu, 2024b; Zhong et al., 2025).

A separate but related direction pursues full-duplex interaction, where models listen and speak simultaneously without explicit turn boundaries (Défossez et al., 2024; Roy et al., 2026; Wang et al., 2025a; Yu et al., 2026; Zhang et al., 2025a). Among these, MiniCPM-o 4.5 (Cui et al., 2026) is especially close in modality coverage, as it supports full-duplex interaction with video and audio input, but it is designed as a general-purpose assistant rather than for egocentric task guidance. Despite this progress, these models are designed for reactive dialog and do not address proactive intervention during ongoing physical tasks.

Egocentric Task Understanding and Datasets. Large-scale egocentric video datasets have enabled research on activity understanding from a first-person perspective. Ego4D (Grauman et al., 2022) established benchmarks for episodic memory, forecasting, and social interaction, and EPIC-KITCHENS (Damen et al., 2018) provided densely annotated cooking sessions for action recognition

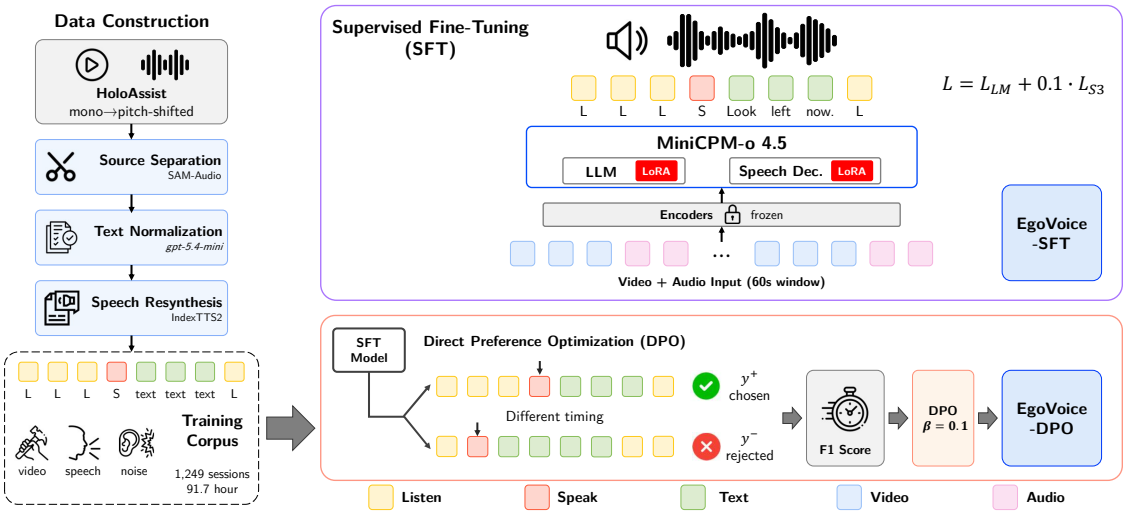


Figure 2: Overview of the EgoVoice framework. We process HoloAssist recordings into clean training data (left, Section 3.1), fine-tune MiniCPM-o 4.5 to learn listen-or-speak behavior (top right, EgoVoice-SFT), then refine intervention timing with DPO (bottom right, EgoVoice-DPO, Section 3.3).

and anticipation. HoloAssist (Wang et al., 2023) captured instructor-guided task assistance sessions with synchronized egocentric video and audio, including annotations for intervention type prediction. Together with Assembly101 (Sener et al., 2022) and Ego-Exo4D (Grauman et al., 2024), these datasets have advanced procedure understanding, but none targets continuous spoken guidance during ongoing tasks. Among these, HoloAssist is closest to our setting as it contains real instructor speech, though its audio is recorded as a mono channel with pitch-shifted speaker identities, requiring additional processing before being used for voice interaction training.

Proactive Assistance in Streaming Video. Proactive assistance has been studied broadly in dialog systems, where models anticipate user needs and initiate responses without explicit prompts (Deng et al., 2023; Wu et al., 2019). This idea has been extended to streaming video LLMs, which introduce per-frame speak-or-stay-silent decisions (Chen et al., 2024; Wang et al., 2025b) and improve timing through reinforcement learning, specialized objectives, or modular architectures (Wang et al., 2026; Qian et al., 2025; Lu et al., 2026; Yan et al., 2026).

In egocentric settings, ProAssist (Zhang et al., 2025c) proposed a data curation pipeline, evaluation protocols, and an end-to-end streaming model for proactive dialog from video, and ESTP (Zhang et al., 2026) introduced a multi-stage training strategy for egocentric streaming video understanding. Prior work has shown that LLMs can generate appropriate content but struggle with the timing of spoken responses (Umair et al., 2024), and EgoSpeak (Kim et al., 2025) studied this problem by training an online prediction model on egocentric visual cues. Mirai (Fang et al., 2025a) and ProMemAssist (Pu et al., 2025) demonstrated system prototypes for proactive spoken assistance on wearable devices, though they rely on cascaded API calls. These methods produce text, predict only when to speak, or rely on cascaded pipelines, rather than training a single model for intervention timing and spoken guidance from egocentric streams.

3 Method

EgoVoice converts egocentric instructional recordings into a format suitable for proactive spoken assistants. Building upon MiniCPM-o 4.5 (Cui et al., 2026), a streaming full-duplex multimodal model, we adapt its listen-or-speak interface to egocentric task assistance through supervised fine-tuning and direct preference optimization. We first describe how we construct the dataset from HoloAssist (Section 3.1), then present the evaluation protocol for measuring proactive spoken assistance (Section 3.2), and finally the training pipeline that learns and refines intervention behavior (Section 3.3). An overview of EgoVoice is in Figure 2.

3.1 Data Construction

Perhaps the most straightforward approach to building an egocentric proactive spoken assistant would be to start from an existing video-text dataset (Zhang et al., 2025c; Wang et al., 2025b) and convert the textual instructions into speech. However, this strategy has fundamental limitations

as illustrated in Figure 4. Text-based benchmarks assume that the full instruction is visible to the user at the moment it is issued. Speech, by contrast, takes time to play back. If a synthesized utterance is placed at the same timestamp as the original text instruction, the user will already be acting on the advice before the utterance finishes playing, creating a mismatch between the audio timeline and the user's behavior. Shifting the synthesis earlier to avoid mismatch is also unreliable, as the user's actions are continuous and the shifted utterance may collide with other instructions or fall during an unrelated action. Moreover, synthesizing speech from text labels does not account for the ambient sounds that naturally accompany real-world physical tasks.

For these reasons, we start from HoloAssist (Wang et al., 2023), a large-scale corpus of egocentric task assistance in which a user performs physical tasks while receiving spoken guidance from a human instructor. HoloAssist's spoken interventions are already aligned with the user's behavior through real-world recording, and preserve background noises. However, the raw recordings are not directly usable for training a multimodal voice interaction model. The video comes with a single audio channel in which user speech, instructor speech, and background sounds are mixed, and speaker identities are anonymized through pitch shifting, which makes them sound unnatural. We therefore process each session as follows.

First, we separate background audio from the mixture. The pitch shifting renders the voices unsuitable for training a natural-sounding voice interaction model, so we isolate only the background sounds. We use SAM-Audio (Shi et al., 2025), a prompt-based source separation model, to extract human voices from the mixture using the speech-containing intervals as prompts and retain the residual as the background channel. We then apply Silero VAD (Silero Team, 2024) to verify that no residual human speech remains in the separated background, and discard all sessions where voice activity is still detected.

Second, we synthesize clean user and instructor speech. The original transcripts in the HoloAssist metadata contain spelling errors, inconsistent punctuation, and disfluencies, which we normalize using *gpt-5.4-mini* (OpenAI, 2026). We also remove task-irrelevant opening and closing remarks (9.0% of utterances). We then resynthesize with IndexTTS2 (Zhou et al., 2026), a speaker-adaptive TTS, with per-session reference speakers sampled from the English subset of Multilingual LibriSpeech (Pratap et al., 2020) (2,498 training speakers, 340 disjoint evaluation speakers). Each utterance is synthesized multiple times; candidates exceeding the original duration are discarded to preserve timing, and the candidate whose ASR transcription exactly matches the original text, as verified by Qwen3-ASR-1.7B (Shi et al., 2026), is selected. Sessions containing any mispronounced utterance are discarded after manual verification. Further details are provided in Appendix A.1.

After this full pipeline, the original HoloAssist corpus of 1,673 instructional sessions (124.8 hours) yields 1,249 training sessions (91.7 hours) and 170 held-out benchmark sessions (12.9 hours). Each processed session contains synchronized egocentric video, resynthesized user speech, resynthesized instructor speech, and separated background audio.

3.2 Evaluation Protocol

We propose an evaluation protocol for measuring proactive spoken assistance over full egocentric task sessions. Following ProAssist (Zhang et al., 2025c), we provide each model with a system prompt containing the task name and step-by-step recipe, so that models have access to the procedural context when deciding whether and what to speak.

Onset Timing. We match predicted utterances to reference instructor utterances using the Hungarian algorithm (Kuhn, 1955) with an asymmetric onset window of -2.5s (early) to $+1.5$s (late), reflecting that early interventions are less disruptive than late ones. A predicted utterance can match a reference only if its onset falls within this window. From the matched pairs, we compute Precision, Recall, and F1 as count-based scores over the number of matched, predicted, and reference utterances. All timing metrics are computed per session and macro-averaged across the 170 benchmark sessions.

End-of-Utterance Timing. Onset metrics do not capture whether the model finishes speaking before the user needs to act. We therefore measure end-vs-action, the gap between the start of the user's next action and the predicted utterance end time. A positive value means the model finished speaking before the user needed to act; a negative value means the model was still talking when the user began the action. Since this metric depends on when an utterance begins and how long it lasts, we also report its decomposition. We define start-vs-action as the gap between the start of the user's next action and the predicted utterance start time; end-vs-action then

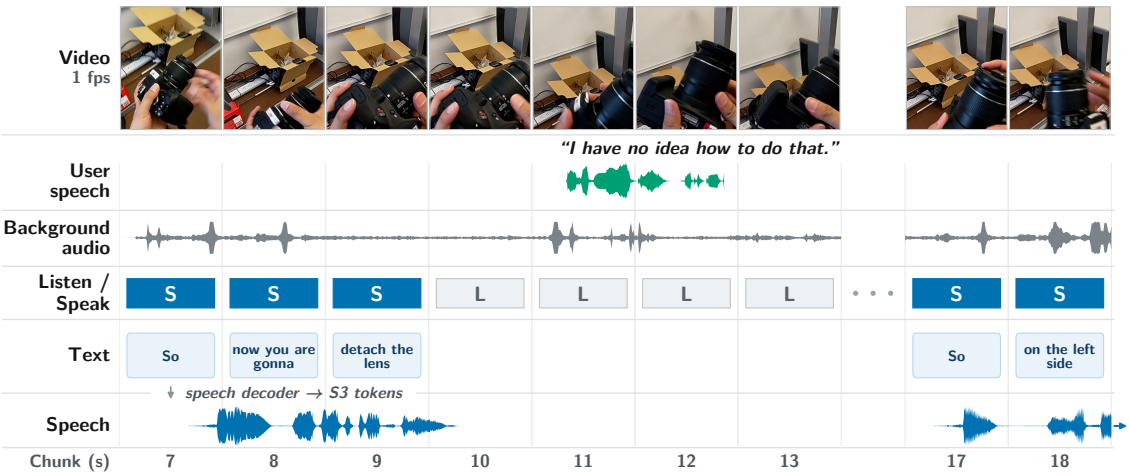


Figure 3: The streaming interface of EgoVoice (`R101-2Aug-DSLR`). Each chunk carries one video frame and one second of audio, and the model emits LISTEN or SPEAK for each. On SPEAK chunks the LLM emits text tokens, and the speech-token decoder turns them into S3 speech tokens that are decoded into the waveform the user hears.

equals start-vs-action minus the utterance duration for every matched pair (Appendix A.4).

Since end-vs-action is meaningful only when the utterance is a procedural directive that prompts a user action rather than a brief acknowledgement, we define an action-linked evaluation subset. Using *gpt-5.5* (OpenAI, 2026), we classify each reference utterance as a procedural directive or non-directive and link directives to the user's nearest subsequent action within T=5 seconds, yielding 942 of 2,067 reference utterances across 170 benchmark sessions. Unlike the other metrics, end-vs-action is reported only on this action-linked subset.

Content Relevance. For each temporally matched pair, we compute cosine similarity between the predicted and reference texts using Qwen3-Embedding-0.6B (Zhang et al., 2025b).

Overall Assistance Quality. We use an LLM-as-a-judge evaluation following the session-level assessment protocol of ProAssist (Zhang et al., 2025c). We prompt *gpt-5.4-mini* (OpenAI, 2026) (reasoning effort=medium) to rate each session on four axes on a 1-to-5 Likert scale: Correctness (whether each utterance is accurate given the reference), Promptness (whether the assistant intervenes at the right moment), Efficiency (whether responses are concise and free of redundancy), and Overall (the overall helpfulness of the assistant's behavior). The judge receives the full session context, with reference and predicted utterances timestamped. Further details are provided in Appendix A.4.

Evaluating Intended and Delivered Speech. Most multimodal speech models produce both a text response and a spoken response. We therefore evaluate all metrics from two sources: `text-llm`, the text that the model generates internally (reported in Appendix A.6.1), and `text-asr`, the text obtained by transcribing the model's spoken output with ASR. The former reflects what the model intended to say; the latter reflects what the user actually hears. For timing, `text-llm` uses the chunk index at which the model produces the text (described in Section 3.3). For `text-asr`, we first transcribe the generated audio with Qwen3-ASR-1.7B, then obtain word-level alignment using Qwen3-ForcedAligner-0.6B (Shi et al., 2026) to determine the precise onset and offset of each utterance.

3.3 Training

We train EgoVoice by fine-tuning MiniCPM-o 4.5 (Cui et al., 2026), a streaming full-duplex multimodal model that processes interleaved video and audio chunk by chunk. At each one-second chunk, the model encodes the current video frame and audio segment (background audio and user speech), appends the resulting tokens to a running context, and produces a decision: either a LISTEN token, indicating that the model should remain silent, or a SPEAK token, after which the LLM backbone autoregressively generates a text response. For each text token, the corresponding LLM hidden state is passed to a speech-token decoder (0.3B parameters), which produces S3 speech tokens (Du et al., 2024) converted to a waveform by a streaming flow-matching decoder (Lipman et al., 2023). Figure 3 shows this process on a benchmark session.

For training, we split each processed session from Section 3.1 into 60-second windows at one-second chunk granularity. Each chunk t within a window is represented as $s_t = (v_t, a_t, z_t, w_t, c_t)$,

where v_t is the egocentric video frame, a_t is the input audio containing background audio and user speech, $z_t \in \{\text{LISTEN}, \text{SPEAK}\}$ is the intervention label, w_t is the normalized text instruction, and c_t is the corresponding S3 speech-token sequence (both empty when $z_t = \text{LISTEN}$). Instructor speech is never included in the model input; it is used exclusively as the target output.

Each instructor utterance is segmented into one-second chunks using word-level timestamps from Qwen3-ForcedAligner-0.6B, distributing the text and speech tokens across the corresponding chunks. For each reference utterance, we construct a 60-second training window centered on that utterance (with slight temporal jitter), so that every window contains at least one intervention event surrounded by listening context. This yields 13,676 training windows per epoch. At evaluation, models process complete sessions without windowing.

Supervised Fine-Tuning (SFT). We first train the model to learn the basic behavior of proactive spoken assistance: remain silent during most chunks, but generate task-relevant spoken guidance when intervention is needed. We apply LoRA (Hu et al., 2022) to both the LLM backbone and the speech-token decoder, while keeping the vision encoder, audio encoder, and other adapters frozen. Within each window, the model processes chunks autoregressively, emitting either a LISTEN token or a SPEAK token followed by the guidance text in byte-pair encoding (BPE) tokens and the S3 speech-token sequence. The training loss is $\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{LM}} + \lambda \cdot \mathcal{L}_{\text{S3}}$ with $\lambda = 0.1$, where $\mathcal{L}_{\text{LM}}$ is a single autoregressive cross-entropy over LISTEN/SPEAK decisions, BPE text tokens, and end-of-turn markers, and $\mathcal{L}_{\text{S3}}$ is a cross-entropy on the speech-token decoder over S3 speech-token frames for SPEAK chunks only. Gradients from $\mathcal{L}_{\text{S3}}$ are stopped at the LLM hidden states, so the two losses do not share backward signal through the backbone.

Direct Preference Optimization (DPO). Supervised fine-tuning provides a strong starting point, but proactive assistance admits multiple acceptable behaviors. We further refine the SFT model with DPO (Rafailov et al., 2023), allowing the model to learn from comparisons between alternative candidate samples under the same streaming context.

From the SFT model, we generate N=10 candidate samples per training window, each conditioned on the same visual and audio context. We score each candidate with the F1 defined in Section 3.2, computed after temporal matching using `text-llm`. We construct preference pairs by selecting the candidate with the highest F1 as the chosen sample y^+ and randomly sampling the rejected sample y^- from the bottom three candidates by F1, requiring a minimum F1 gap of 0.10. This yields 5,335 preference pairs from windows across 1,021 training sessions.

We train with length-normalized DPO (Lambert et al., 2025). Given a streaming context x and a preference pair (y^+, y^-), the objective is:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}\left[\log \sigma\left(\frac{\beta}{|y^+|}\rho_{y^+} - \frac{\beta}{|y^-|}\rho_{y^-}\right)\right], \tag{1}$$

where $\rho_y = \log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x)$ and β is a hyperparameter controlling the strength of the preference constraint. Here, y denotes the sequence of LISTEN/SPEAK decisions and generated text tokens over the window; S3 speech tokens are excluded from the DPO likelihood. The reference model π_{ref} is the frozen SFT model. For DPO, we apply LoRA to the LLM backbone only, since preferences are constructed over intervention decisions and textual guidance rather than speech synthesis.

4 Experiments

Baselines. We compare EgoVoice against six baselines spanning three categories. For closed-source APIs, we evaluate Gemini Live (`gemini-live-2.5-flash-native-audio`) and two snapshots of the OpenAI Realtime API (`gpt-realtime` and `gpt-realtime-2`), all of which accept streaming video and audio and produce both text and speech. Since these APIs rely on voice activity detection and rarely respond without active user speech, we remove background audio to prevent false triggering, providing a more favorable evaluation condition.

As a reference open-source text baseline, we evaluate ProAssist (Zhang et al., 2025c) (8B), the most directly comparable prior work, which uses the HoloAssist data and evaluation protocol but handles text rather than speech. Following the ProAssist protocol, it receives streaming video together with ground-truth transcripts of the user's speech at their original timestamps, and a neutral opening query (e.g., "help me assemble this lamp") for sessions in which the user never speaks.

We also evaluate two open-source baselines that support voice interaction. The cascaded baseline (+ ASR/TTS cascade in Table 1) pairs ProAssist with Qwen3-ASR-1.7B (Shi et al., 2026) for the user's

Model	Onset Timing			Content	LLM Judge (1–5)				End Timing
	Prec.	Rec.	F1	Cosine	Correct.	Prompt.	Effic.	Overall	end-vs-act (s)
Gemini Live	.379	.019	.022	.448	2.09	2.39	2.67	1.94	−3.20 / −2.26
`gpt-realtime`	.285	.024	.030	.500	2.14	2.60	2.52	2.10	−7.40 / −7.46
`gpt-realtime-2`	.256	.019	.024	.456	2.28	2.40	2.10	2.14	−16.1 / −16.8
ProAssist†	.480	.158	.195	.594	2.34	2.63	**2.99**	2.06	–
+ ASR/TTS cascade	.479	.158	.195	.564	2.13	2.50	2.87	1.92	−0.44 / −1.56
MiniCPM-o 4.5	.259	.108	.113	.519	1.54	1.63	1.73	1.41	−4.69 / −0.40
EgoVoice-SFT	.467	.235	.274	**.684**	**2.41**	**2.64**	2.64	**2.23**	+1.57 / +1.52
EgoVoice-DPO	**.482**	**.285**	**.324**	.649	2.35	2.55	2.42	2.18	**+1.75 / +1.62**
Human instructor‡	–	–	–	–	–	–	–	–	+1.69 / +1.38

Table 1: Results on 170 benchmark sessions (`text-asr`). † indicates a text-only baseline, evaluated on `text-llm`. ‡ indicates the human instructor, whose utterances are the reference for all systems. End-vs-act is reported as mean / median on the action-linked subset (T=5), where positive values mean the model typically finishes speaking before the user acts; per-intervention distributions are in Appendix A.6.2. Best in **bold**, second best underlined.

speech and IndexTTS2 (Zhou et al., 2026) for the assistant's speech, using the same per-session reference speaker as our benchmark. We assume ideal zero latency between ASR, ProAssist, and TTS, so that each utterance begins playing at the chunk where ProAssist generates it and only the duration of the synthesized speech affects when it ends. The end-to-end baseline is MiniCPM-o 4.5 (Cui et al., 2026) (9B) in a zero-shot setting, which also serves as EgoVoice's backbone before fine-tuning.

Our Models. We evaluate EgoVoice-SFT, trained with SFT on our processed data, and EgoVoice-DPO, which further refines EgoVoice-SFT with DPO on intervention timing.

Implementation Details. For SFT, we apply LoRA (r=32, α=64) to both the LLM backbone and the speech-token decoder, training for 5 epochs on 13,676 windows with a learning rate of 2e-4 and a batch size of 32. For DPO, we train EgoVoice-SFT with β=0.1 for 1 epoch with a learning rate of 5e-6 and a batch size of 16, and select the checkpoint based on validation performance. Additional details are provided in Appendices A.2 and A.3.

5 Results

We evaluate all models on the 170 benchmark sessions. Unless otherwise noted, we report metrics on the `text-asr` channel, which reflects what the user actually hears; `text-llm` results and qualitative examples are provided in Appendix A.6. Reported p-values are from a paired Wilcoxon signed-rank test (Wilcoxon, 1945) over sessions.

5.1 Model Comparison

Table 1 shows that off-the-shelf models struggle to produce well-timed proactive interventions. The closed-source APIs all achieve F1 scores of at most 0.03, as they rely on server-side VAD that triggers responses only when the user speaks; since users are largely silent during physical tasks, these models rarely find an occasion to intervene. MiniCPM-o 4.5 in a zero-shot setting performs better (F1 = 0.113) by virtue of its listen-or-speak mechanism, but its judge scores are the lowest overall (Overall = 1.41), reflecting a tendency to narrate observed actions rather than provide task-relevant guidance. The same behavior shows up in its end-vs-action, where a tail of very long narrations pulls the mean far below the median (−4.69s against −0.40s).

EgoVoice-SFT substantially improves over MiniCPM-o 4.5 across all metrics, with $p < 10^{-12}$ on both onset F1 and LLM Judge (Overall). EgoVoice-SFT also outperforms ProAssist on F1 score (0.274 vs 0.195, $p < 10^{-4}$) and cosine similarity (0.684 vs 0.594), even though ProAssist is a text-only model evaluated using `text-llm`. Nevertheless, ProAssist achieves the highest Efficiency score among all models. It speaks less often than EgoVoice and in longer turns (2.2 utterances of 16.6 words per session, against 5.6 utterances of 7.8 words), with almost no brief acknowledgements: utterances of three words or fewer make up 0.8% of its output, against 19.3% for EgoVoice-DPO and 17.9% for the instructor. The Efficiency axis appears to favor this turn structure (Appendix A.6.3).

Turning ProAssist into a spoken assistant (+ ASR/TTS cascade in Table 1) does not close the gap: because the cascade inherits ProAssist's intervention decisions, its onset F1 is unchanged at 0.195. In addition, the cascade's speech ends after the user has already begun the linked action in 62.0% of matched interventions, against 24.3%

Variant	Onset Timing			Content	LLM Judge (1–5)				End Timing
	Prec.	Rec.	F1	Cosine	Correct.	Prompt.	Effic.	Overall	end-vs-act (s)
SFT design choices									
EgoVoice-SFT	.467	.235	.274	.684	2.41	2.64	2.64	2.23	+1.57 / +1.52
w/o recipe	.398	.223	.240	.621	1.91	2.14	2.15	1.81	+2.32 / +2.09
w/o detach	.435	.246	.276	.680	2.38	2.61	2.45	2.18	+2.19 / +1.86
DPO variants (all from EgoVoice-SFT)									
EgoVoice-DPO	.482	.285	.324	.649	2.35	2.55	2.42	2.18	+1.75 / +1.62
Vanilla DPO, β=0.1	.458	.261	.298	.651	2.37	2.70	2.58	2.25	+1.14 / +1.40
SimPO	.457	.309	.328	.641	2.38	2.60	2.40	2.18	+1.86 / +1.77
Length-norm., β=0.5	.401	.371	.345	.617	2.22	2.51	2.02	2.04	+1.62 / +1.62
F1 + Cosine reward	.451	.205	.248	.674	2.36	2.61	2.60	2.16	+1.51 / +1.53

Table 2: Ablation results (`text-asr`). Metrics are computed following the same protocol as Table 1. EgoVoice-DPO uses length-normalized DPO with $\beta = 0.1$. Each variant differs from this reference in only the listed factor.

for EgoVoice-SFT and 28.2% for EgoVoice-DPO, giving it negative mean and median end-vs-action values. Deciding when to speak is therefore not sufficient on its own, because the length of the resulting speech determines whether the guidance arrives before it is needed.

EgoVoice-SFT and EgoVoice-DPO are the only models with positive end-vs-action by both mean and median, and their margins (+1.57s and +1.75s) are close to the instructor's own (+1.69s). Between the two, EgoVoice-DPO further improves onset timing, raising F1 by 0.050 (p = 0.0007) as both precision and recall increase, indicating that DPO encourages the model to intervene at more of the right moments rather than simply producing more speech. This comes at a small cost in content similarity (0.684 to 0.649), while LLM Judge (Overall) does not separate the two variants (95% bootstrap CI $[-0.224, 0.072]$, p = 0.42). The two stages are thus complementary: EgoVoice-DPO prioritizes intervention timing, while EgoVoice-SFT retains a small edge on content metrics.

5.2 Ablation Studies

Table 2 examines individual design choices. For SFT, removing the recipe from the system prompt so that the model relies solely on video and audio causes the largest degradation across all metrics except end-vs-action. This confirms that procedural context is essential for generating relevant guidance, not just for what to say but also for when to say it. We also examine the effect of gradient detachment between the LLM backbone and the speech-token decoder. Removing detachment yields slightly higher F1 but lower judge scores, suggesting that speech loss gradients flowing into the LLM backbone interfere with content quality.

For DPO, we first compare three loss variants: length-normalized DPO (Lambert et al., 2025), which normalizes by sequence length; vanilla DPO (Rafailov et al., 2023); and SimPO (Meng et al., 2024), a reference-free variant. All three improve F1 over SFT, with modest differences among them: trade-offs across cosine similarity and judge scores rather than clear wins. We select length-normalized DPO as our default because it provides the most balanced performance across timing, content, and end-vs-action metrics.

We then examine the sensitivity to β and the reward strategy. Increasing β beyond 0.10 leads the model to speak more often without meaningful content, which raises recall and superficially improves F1 but degrades precision and other metrics related to response quality. To address the content quality decline, we also construct preference pairs requiring improvement on both F1 and cosine similarity. However, this composite strategy instead leads to degradation in both metrics relative to SFT. Selecting by F1 alone proves more effective, suggesting that a single clear optimization signal is preferable to a composite one in this setting.

5.3 Human Evaluation

We additionally run pairwise human preference evaluations on Amazon Mechanical Turk. Each evaluator watches two versions of the same egocentric clip, differing only in which assistant's spoken response is overlaid in the clip, and selects the preferred assistant or a tie. We recruit 100 workers per comparison and discard those who fail a built-in attention check, retaining 252–267 valid judgments per comparison. The full protocol and the evaluation template are in Appendix A.5.

As shown in Table 3, both EgoVoice-SFT and EgoVoice-DPO are preferred over MiniCPM-o 4.5 ($p < 0.05$), confirming that the improvements ob-

Comparison	Win	Tie	Lose	p-val
SFT vs MiniCPM	**45.6%**	23.0%	31.3%	0.01
DPO vs MiniCPM	**45.3%**	21.7%	33.0%	0.02
DPO vs SFT	**36.1%**	30.6%	33.3%	0.60

Table 3: Pairwise human preference evaluation results.

served in automatic metrics translate to perceptible gains in human judgment. Evaluators show a slight numerical preference for DPO over SFT, but the difference is not significant ($p = 0.60$), suggesting that the timing gains from DPO and the content advantage of SFT are perceived as comparably valuable.

5.4 Analysis

Intervention Ordering. Onset F1 measures whether interventions land at the right moments, but not whether they follow the order of the task, since the Hungarian matching pairs each prediction with a reference independently of every other pair. A model could therefore match many references while explaining a later step before an earlier one. We test this with Kendall's τ_b, a rank correlation that counts how often two orderings agree on which of a pair of items comes first, ranging from -1 for a reversed order to 1 for an identical one, with a correction for tied ranks (Kendall, 1945). We compute it per session between the order in which the model produced its matched interventions and the order of the references they matched. Over sessions with at least two matched pairs, EgoVoice-DPO reaches 0.868 and EgoVoice-SFT 0.841, against 0.850 for ProAssist and 0.749 for MiniCPM-o 4.5. The trained models thus move through the task in close to the order the instructor did, while the zero-shot backbone is markedly less consistent.

Content Alignment. A model could deliver guidance in the right order yet phrase it generically enough to fit any nearby step, which onset matching alone would not detect. For each matched pair we therefore compare the prediction against its matched reference and the two references before and the two after it, using the same cosine similarity as our Content Relevance metric, and record which is closest. The matched reference is closest for 59.5% of EgoVoice-SFT pairs and 52.8% of EgoVoice-DPO, against 45.9% for ProAssist and 31.8% for MiniCPM-o 4.5. All systems thus carry some step-specific content, but the trained models are the most locally precise.

Utterance Types. Neither timing nor content metrics say what kind of speech a model produces. We classify every delivered utterance, and every reference utterance from the human instructors, into proactive guidance, third-person scene narration, and other using *gpt-5.6-luna* (Table 4; the prompt and protocol are in Appendix A.4). Our models deliver guidance more often than the backbone and bring narration close to the instructors' level. What remains is the other category, which for every system consists mostly of brief acknowledgments: 70.7% of EgoVoice-DPO's are three words or fewer. EgoVoice-SFT and EgoVoice-DPO generate more of them than the instructors, and that gap accounts for most of the remaining distance to the reference distribution.

Model	Guidance	Narration	Other
Human instructor	76.7	5.6	17.8
MiniCPM-o 4.5	30.1	21.0	49.0
EgoVoice-SFT	65.7	5.4	29.0
EgoVoice-DPO	64.2	5.5	30.3

Table 4: Distribution of utterance types on the 170 benchmark sessions (%). Each utterance is classified as proactive guidance, third-person scene narration, or other. The first row is the reference instructor speech.

Failure Modes. Quantitative results in Table 1 hide two failure modes. First, a positive mean end-vs-action does not guarantee the model finishes in time: EgoVoice-DPO still finishes after the user has begun the linked action in 28.2% of matched interventions and EgoVoice-SFT in 24.3%, which is well below MiniCPM-o 4.5 (54.5%) and the cascade (62.0%), but far from solved. Second, recall is the binding constraint: at a recall of 0.285, EgoVoice-DPO misses most moments where the instructor intervened. Appendix A.6.5 groups examples of each by failure type.

6 Conclusion

We presented EgoVoice, a framework for training and evaluating proactive egocentric spoken assistants. We constructed a spoken interaction dataset from HoloAssist through source separation and speech resynthesis, and proposed an evaluation protocol covering intervention timing, content relevance, and overall assistance quality. Through supervised fine-tuning and direct preference optimization on MiniCPM-o 4.5, we showed that both EgoVoice-SFT and EgoVoice-DPO outperform all baselines on onset F1 and content relevance. We hope that EgoVoice serves as an initial step toward building always-on spoken assistants that can proactively guide users through real-world tasks.

Limitations

While we demonstrated the potential of egocentric proactive spoken assistance, our framework has several limitations. First, from a data perspective, our dataset is constructed from HoloAssist, which covers a specific set of procedural assembly tasks. As a result, the model's ability to generalize to other domains such as cooking, outdoor navigation, or social interaction is not guaranteed. In addition, there is still a substantial performance gap between our model and the ground truth instructor behavior (Appendix A.6.2), suggesting that more diverse training data and larger-scale training would be needed to approach human-level proactive assistance.

Second, our backbone model, MiniCPM-o 4.5, occasionally exhibits several issues that we observed during our experiments, including excessively long responses, severe repetition, and occasional language switching (e.g., generating Chinese tokens or unrecognizable special characters in the middle of English utterances). Although these artifacts are reduced through our fine-tuning, they still carry over into both EgoVoice-SFT and EgoVoice-DPO, as LoRA adapts only a small subset of model parameters. The model also inherits the one-second chunk granularity of MiniCPM-o 4.5, which may limit the precision of intervention timing in scenarios that require sub-second responsiveness.

Third, our evaluation is conducted on prerecorded sessions: the user's actions are already fixed, so every model is scored against behavior that is independent of its own advice. Our numbers should therefore be read as a proxy for interactive performance rather than a measurement of it. We also do not model overlap between the two speakers: each utterance plays to completion, so the protocol covers neither a user interrupting the assistant nor the assistant yielding mid-utterance. Handling barge-in in both directions is necessary for deployment and is left to future work.

Fourth, our DPO stage improves intervention timing but introduces a slight decline in content quality, indicating that the reward signal has room for further refinement. Finally, deploying an always-on spoken assistant in real-world settings raises privacy risks: continuous egocentric recording may capture bystanders without consent, and transmitting audio and video to external servers risks exposing sensitive information. Mitigating these risks will require bystander consent mechanisms, on-device processing to minimize data transmission, and audio redaction protocols. Nevertheless, with growing industry investment in AR and smart glasses, we believe open-source contributions toward proactive spoken assistance will become increasingly valuable, and we hope our dataset, code, and models will serve as a useful foundation for this emerging direction.

Ethics Statement

Our dataset is derived from HoloAssist (Wang et al., 2023), released under CDLAv2, which permits redistribution and derivative works, and we use all artifacts consistently with their intended use (Appendix A.7). The original recordings anonymize speaker identities by pitch shifting, and our resynthesis replaces every voice with zero-shot TTS output from reference speakers in Multilingual LibriSpeech (Pratap et al., 2020) (CC BY 4.0), a corpus built from public-domain LibriVox audiobook recordings. Both corpora identify speakers only by numeric identifiers; the speech we release is entirely synthetic, and the separated background channel is verified to contain no residual speech (Section 3.1). No voice in our data can therefore be linked to a named person, and we will preserve this property in any future version of the dataset.

For the human evaluation, we recruited workers on Amazon Mechanical Turk, compensated them for each completed task, and collected no personally identifying information beyond what the platform requires; the protocol is described in Appendix A.5. Beyond the study itself, an always-on assistant that continuously records first-person video and audio carries risks for the user and for bystanders. We discuss these risks and possible mitigations in the Limitations section, and we release our models for research on proactive assistance rather than for deployment as a monitoring tool.

Acknowledgments

This work was supported by the Advanced GPU Utilization Support Program funded by the Government of the Republic of Korea (Ministry of Science and ICT), by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. RS-2022-NR068754), by the NVIDIA Academic Grant Program using NVIDIA RTX PRO 6000 GPUs and CUDA, and by the supercomputing resources provided by the Urban Big Data and AI Institute of the University of Seoul

(http://ubai.uos.ac.kr).

References

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 (CVPR)*, pages 18407–18418.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Jiancheng Gui, Luoyuan Zhang, Xian Sun, Fuwei Huang, Moye Chen, Zhuo Lin, Hanyu Liu, Qingxin Gui, Qingzhe Han, and 17 others. 2026. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *Preprint*, arXiv:2604.27393.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023. A survey on proactive dialogue systems: Problems, methods, and prospects. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6583–6591. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, Zhifu Gao, and Zhijie Yan. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *Preprint*, arXiv:2407.05407.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *Preprint*, arXiv:2410.00037.

Cathy Mengying Fang, Yasith Samaradivakara, Pattie Maes, and Suranga Nanayakkara. 2025a. Mirai: A wearable proactive ai "inner-voice" for contextual nudging. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '25, New York, NY, USA. Association for Computing Machinery.

Qingkai Fang, Yan Zhou, Shoutao Guo, Shaolei Zhang, and Yang Feng. 2025b. LLaMA-omni 2: LLM-based real-time spoken chatbot with autoregressive streaming speech synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18617–18629, Vienna, Austria. Association for Computational Linguistics.

Chaoyou Fu, Haojia Lin, Xiong Wang, YiFan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, Long MA, Xiawu Zheng, Rongrong Ji, Xing Sun, Caifeng Shan, and Ran He. 2026. VITA-1.5: Towards GPT-4o level real-time vision and speech interaction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, and 66 others. 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 (CVPR)*, pages 18995–19012.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, and 81 others. 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 (CVPR)*, pages 19383–19400.

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

M. G. Kendall. 1945. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251.

Heeseung Kim, Soonshin Seo, Kyeongseok Jeong, Ohsung Kwon, Soyoon Kim, Jungwhan Kim, Jaehong Lee, Eunwoo Song, Myungwoo Oh, Jung-Woo Ha, Sungroh Yoon, and Kang Min Yoo. 2024. Paralinguistics-aware speech-empowered large language models for natural conversation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Junhyeok Kim, Min Soo Kim, Jiwan Chung, Jungbin Cho, Jisoo Kim, Sungwoong Kim, Gyeongbo Sim, and Youngjae Yu. 2025. EgoSpeak: Learning when to speak for egocentric conversational agents in the wild. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2990–3005,

Albuquerque, New Mexico. Association for Computational Linguistics.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*.

Guan-Ting Lin, Cheng-Han Chiang, and Hung-yi Lee. 2024. Advancing large language models to capture varied speaking styles and respond properly in spoken conversations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6626–6642, Bangkok, Thailand. Association for Computational Linguistics.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

Zuwei Long, Yunhang Shen, Chaoyou Fu, Heting Gao, lijiang Li, Peixian Chen, Mengdan Zhang, Hang Shao, Jian Li, Jinlong Peng, Haoyu Cao, Ke Li, Rongrong Ji, and Xing Sun. 2026. VITA-audio: Fast interleaved audio-text token generation for efficient large speech-language model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, Rui Liu, and Hongsheng Li. 2026. Aura: Always-on understanding and real-time assistance via video streams. *Preprint*, arXiv:2604.04184.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, pages 124198–124235. Curran Associates, Inc.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2026. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. MLS: A Large-Scale Multilingual Dataset for Speech Research. In *Interspeech 2020*, pages 2757–2761.

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

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. 2025. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24045–24055.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rajarshi Roy, Jonathan Raiman, Sang gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. 2026. Personaplex: Voice and role control for full duplex conversational speech models. *Preprint*, arXiv:2602.06053.

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 (CVPR)*, pages 21096–21106.

Bowen Shi, Andros Tjandra, John Hoffman, Helin Wang, Yi-Chiao Wu, Luya Gao, Julius Richter, Matt Le, Apoorv Vyas, Sanyuan Chen, Christoph Feichtenhofer, Piotr Dollár, Wei-Ning Hsu, and Ann Lee. 2025. Sam audio: Segment anything in audio. *Preprint*, arXiv:2512.18099.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-asr technical report. *Preprint*, arXiv:2601.21337.

Silero Team. 2024. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. `https://github.com/snakers4/silero-vad`.

Muhammad Umair, Vasanth Sarathy, and Jan Ruiter. 2024. Large language models know what to say but not when to speak. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15503–15514, Miami, Florida, USA. Association for Computational Linguistics.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi,

and Marc Pollefeys. 2023. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20270–20281.

Xiong Wang, Yangze Li, Chaoyou Fu, Yike Zhang, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long MA. 2025a. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen LLM. In *Forty-second International Conference on Machine Learning*.

Yueqian Wang, Songxiang Liu, Disong Wang, Nuo Xu, Wan Guanglu, Huishuai Zhang, and Dongyan Zhao. 2026. MMDuet2: Enhancing proactive interaction of video MLLMs with multi-turn reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.

Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. 2025b. VideoLLM knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6338–6359, Suzhou, China. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy. Association for Computational Linguistics.

Zhifei Xie and Changqiao Wu. 2024a. Mini-omni: Language models can hear, talk while thinking in streaming. *Preprint*, arXiv:2408.16725.

Zhifei Xie and Changqiao Wu. 2024b. Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities. *Preprint*, arXiv:2410.11190.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025. Qwen3-omni technical report. *Preprint*, arXiv:2509.17765.

Weicai Yan, Yuhong Dai, Qi Ran, Haodong Li, Wang Lin, Hao Liao, Xing Xie, Tao Jin, and Jianxun Lian. 2026. Proact-vl: A proactive videollm for real-time ai companions. *Preprint*, arXiv:2603.03447.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. 2026. SALMONN-omni: A standalone speech LLM without codec injection for full-duplex conversation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *Preprint*, arXiv:2412.02612.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773, Singapore. Association for Computational Linguistics.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chao-Hong Tan, Zhihao Du, and ShiLiang Zhang. 2025a. OmniFlatten: An end-to-end GPT model for seamless voice conversation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14570–14580, Vienna, Austria. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Yichi Zhang, Xin Luna Dong, Zhaojiang Lin, Andrea Madotto, Anuj Kumar, Babak Damavandi, Joyce Chai, and Seungwhan Moon. 2025c. Proactive assistant dialogue generation from streaming egocentric videos. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12044–12068, Suzhou, China. Association for Computational Linguistics.

Yulin Zhang, Cheng Shi, Yang Wang, and Sibei Yang. 2026. Eyes wide open: Ego proactive video-LLM for streaming video. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhisheng Zhong, Chengyao Wang, Yuqi Liu, Senqiao Yang, Longxiang Tang, Yuechen Zhang, Jingyao Li, Tianyuan Qu, Yanwei Li, Yukang Chen, Shaozuo Yu, Sitong Wu, Eric Lo, Shu Liu, and Jiaya Jia. 2025. Lyra: An efficient and speech-centric framework for omni-cognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3694–3704.

Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. 2026. Indextts2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(41):35139–35148.

A Appendix

A.1 Data Construction Details

Source Separation. The original HoloAssist recordings contain a single mono audio channel in which user speech, instructor speech, and background sounds are mixed together. Since all voices are pitch-shifted for anonymization, they cannot be used directly for training a natural-sounding spoken assistant. We therefore isolate only the background sounds using SAM-Audio (large-tv variant[2]), a prompt-based source separation model that takes an audio mixture and a reference audio prompt, extracts the sound matching the prompt, and returns both the extracted target and its residual.

Since our goal is to preserve the background sounds while removing all human voices, we provide the speech segments as prompts so that SAM-Audio extracts and discards them, and we retain the residual as the background channel. For each speech-containing interval identified from the HoloAssist metadata timestamps, we supply the raw voice segment with a ± 0.2 second margin as the conditioning region to avoid sharp boundary artifacts in the separated output. To verify that no human voice remains in the resulting background, we apply Silero VAD to the separated background and discard all sessions where residual speech is still detected.

Text Normalization. To synthesize user and assistant speech that is free of pitch shifting, preserves speaker anonymity, and is separated into distinct audio channels suitable for training, we first normalize the written-form dialog transcripts available in HoloAssist's metadata using a two-stage LLM pipeline with *gpt-5.4-mini*. In Stage A, a generator model receives the full conversation session and applies a text normalization covering spelling correction, punctuation standardization, canonical filler-word handling, and proper-noun formatting. The original HoloAssist conversations also contain task-irrelevant opening and closing remarks related to starting or stopping the recording (e.g., "You can start recording now" or "We are done, you can stop"). These are identified using per-utterance metadata flags from the original annotations and removed, accounting for 9.0% of all utterances. In Stage B, an independent reviewer model (*gpt-5.4-mini*) re-checks the Stage A output for faithfulness and consistency.

[2] https://huggingface.co/facebook/sam-audio-large-tv

Stage	Sessions
Original HoloAssist (train + val)	1,673
After source separation (VAD filter)	1,532
After TTS/WER filter + duration ($\geq$60s)	1,419
Final train split	1,249
Final eval split	170

Table 5: Dataset attrition through the construction pipeline. Filters: (i) Silero VAD with `leak_final=0` to remove source-separation residue, (ii) TTS/WER quality + minimum duration ($\geq$60s). The eval split consists of the surviving sessions from the original HoloAssist evaluation split and is not further filtered.

Speech Resynthesis. With the transcripts normalized for synthesis, we resynthesize clean speech using IndexTTS2 (Zhou et al., 2026), a speaker-adaptive zero-shot TTS model. For each session, we assign one reference speaker to the user and one to the instructor, drawn from the English subset of Multilingual LibriSpeech (Pratap et al., 2020) (2,498 training speakers, 340 disjoint evaluation speakers). Each utterance is synthesized 20 times, and candidates longer than the original utterance duration are discarded to prevent the synthesized speech from extending into the time window where the user performs the next physical action. From the remaining candidates, we transcribe each with Qwen3-ASR-1.7B (Shi et al., 2026) and select the one with the lowest word error rate (WER) against the normalized transcript. Sessions whose selected utterances all achieve WER = 0.0% are automatically retained; all others are manually checked, and any session containing even one incorrectly pronounced utterance is discarded.

Dataset Attrition. Table 5 summarizes the number of sessions retained at each stage of the data construction pipeline. The largest reduction occurs at the source separation stage, where 141 sessions are discarded due to residual speech detected in the separated background.

A.2 Baseline Configurations

Gemini Live API. The Gemini Live API is Google's low-latency real-time API for streaming audio, video frames, and text, generating spoken responses. We use `gemini-live-2.5-flash-native-audio`[3] through Vertex AI with JPEG video frames at 1 fps resized to 768×768, context-window compression

[3] https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/gemini/2-5-flash-live-api

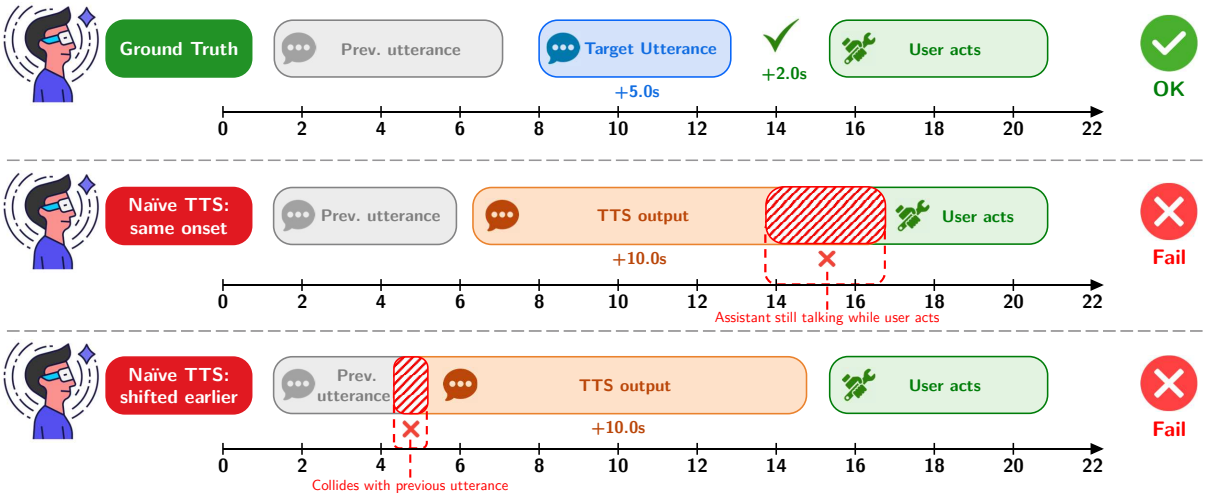


Figure 4: Why naive TTS synthesis on video-text dialog data is insufficient. The ground truth instructor (top) times utterances to finish before the user's next action. Naive TTS (middle) generates longer speech that overlaps with the user's action. Shifting the onset earlier (bottom) avoids the action overlap but collides with the previous utterance and delivers guidance before the user has seen the relevant context. This motivates our choice to start from HoloAssist, where real instructor speech timing is already aligned with user actions, and to resynthesize speech with a strict duration constraint that preserves this alignment.

and proactive audio enabled, and a 15-second tail wait after all video frames are sent to allow the model to finish generating.

OpenAI Realtime API. The OpenAI Realtime API enables low-latency speech-to-speech interaction with GPT-class models over WebSocket. We evaluate two snapshots, `gpt-realtime`[4] and `gpt-realtime-2`[5], with the `alloy` voice, server-side VAD for turn detection, and video frames at 1 fps (768 px). Input audio is 16-bit PCM at 24 kHz.

Both the Gemini Live API and OpenAI Realtime API rely on voice activity detection to trigger responses, meaning they respond primarily when the user speaks. Since users are largely silent during physical tasks in our setting, these models rarely intervene: the average number of utterances per session is 0.25 for Gemini Live, 0.79 for `gpt-realtime-2`, and 0.94 for `gpt-realtime`. We also remove background noise from the input audio for these models to prevent false VAD triggering, providing a more favorable evaluation condition, though the results suggest that these models are not designed for proactive assistance in this setting.

ProAssist. ProAssist (Zhang et al., 2025c) is an 8B open-source video-text model built on VideoLLM-Online (Chen et al., 2024) that learns proactive dialog generation from streaming egocentric video. It handles text rather than speech and does not accept audio input. We use the default settings from the original paper and official implementation, with native 2 fps video at 384×384 resolution. In the cascade variant, the ground-truth user transcripts are replaced by Qwen3-ASR-1.7B transcripts of the resynthesized user speech, and ProAssist's text output is synthesized with IndexTTS2.

MiniCPM-o 4.5. MiniCPM-o 4.5 (Cui et al., 2026) is a 9B open-source omni-modal model that supports real-time full-duplex interaction across vision, speech, and text. It serves as our backbone before fine-tuning. We evaluate it in a zero-shot setting with a system prompt that provides the task name and step-by-step procedural instructions, using temperature = 0.7, top-p = 0.8, and top-k = 100. Unlike the closed-source APIs that rarely speak, MiniCPM-o 4.5 generates an average of 4.85 utterances per session. However, an analysis of all 825 utterances across the 170 benchmark sessions reveals that only 30.1% constitute proactive guidance (Table 4). The remaining utterances are third-person scene narration (21.0%, e.g., "The person is unscrewing the leg"), which is inappropriate for a first-person assistant, or acknowledgements, greetings, and hallucinated text (49.0%, e.g., "Is Jesus Christ", "Oh", or garbled tokens).

A.3 Training Hyperparameters

We provide the full hyperparameter settings for SFT and DPO training in Table 6. All results re-

[4] https://developers.openai.com/api/docs/models/gpt-realtime

[5] https://developers.openai.com/api/docs/models/gpt-realtime-2

	Hyperparameter	SFT	DPO
LoRA	Rank (r) / Alpha (α) / Dropout	32 / 64 / 0.05	
	Target modules	All linear (LLM + TTS)	All linear (LLM only)
	TTS decoder	Trained	Frozen
Loss	Objective	$\mathcal{L}_{\text{LM}} + \lambda \mathcal{L}_{\text{S3}}$	Length-norm. DPO
	Key parameter	$\lambda = 0.1$, detach = True	$\beta = 0.1$
Optimization	Learning rate	2e-4	5e-6
	Schedule	Cosine with 5% warmup	
	Weight decay	0.01	0.0
	Max gradient norm	1.0	
	Effective batch size	32	16
Training	Epochs / steps	5 / 2,140	1 / 334
	Data per epoch	13,676 windows	5,335 pairs
Infra.	Precision	bf16 + FlashAttention-2 (Dao, 2024)	

Table 6: Training hyperparameters for SFT and DPO.

ported in this paper are from single training runs. Training and evaluation are performed using three GPU types: NVIDIA A6000, RTX PRO 6000, and B200.

A.4 Evaluation Details

LLM Judge. We prompt *gpt-5.4-mini*[6] (reasoning effort = medium) with the full session context, including all reference and predicted utterances with their timestamps. Our judge prompt and scoring rubric are largely adapted from ProAssist (Zhang et al., 2025c). The judge rates each session on four axes on a 1-to-5 Likert scale: Correctness, Promptness, Efficiency, and Overall. The full system prompt is shown in Figure 5.

Action-Linked Classification. To construct the action-linked evaluation subset for end-vs-action timing, we classify each reference utterance as either a procedural directive (e.g., "change the battery") or a non-directive remark (e.g., acknowledgement, encouragement) using *gpt-5.5*[7] (reasoning effort = high) with 5-way self-consistency majority voting at temperature = 1.0. Procedural directives are linked to the user's nearest subsequent physical action from HoloAssist's action annotations within a temporal window of T = 5 seconds. This retains 942 out of 2,067 total reference utterances across 170 benchmark sessions. The classification prompt is shown in Figure 6.

Utterance Type Classification. To characterize what kind of speech each system generates (Table 4), we classify every delivered utterance, and every reference instructor utterance, as proactive guidance, third-person scene narration, or other. We use *gpt-5.6-luna*[8] with the prompt in Figure 7, taking the strict majority over 5 independent calls. The same prompt is applied to model output and to reference utterances so that the two are directly comparable. Across the four sources, 91.6% to 96.3% of utterances received a unanimous label.

Content Similarity. We compute cosine similarity between reference and predicted utterances using Qwen3-Embedding-0.6B (Zhang et al., 2025b). For each matched reference-prediction pair identified by the onset timing metric, we encode both utterances and compute their cosine similarity as a measure of semantic alignment.

ASR Transcription. For the `text-asr` evaluation channel, we transcribe the generated speech using Qwen3-ASR-1.7B (Shi et al., 2026). To determine utterance boundaries within continuous speech output, we apply Qwen3-ForcedAligner-0.6B (Shi et al., 2026) for word-level timestamp alignment, followed by Silero VAD (threshold = 0.5) (Silero Team, 2024) for utterance segmentation. All artifacts and their licenses are listed in Table 11 (Appendix A.7).

End-of-Utterance Decomposition. On the action-linked subset, every matched prediction has a user action attached to it through the reference utterance it matched. We measure three things against the start of that action: when the predicted utterance begins (start-vs-action), when it ends (end-vs-action), and how long it lasts. Both the onset and the end are read from the generated audio: we anchor the utterance at the chunk in which the model starts speaking and locate its first and last

[6] https://developers.openai.com/api/docs/models/gpt-5.4-mini

[7] https://developers.openai.com/api/docs/models/gpt-5.5

[8] https://developers.openai.com/api/docs/models/gpt-5.6-luna

LLM Judge System Prompt (*gpt-5.4-mini*, reasoning effort = medium)

```
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 your analysis concise and to the
point.
2. Evaluate the following aspects:
- 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
```

Figure 5: System prompt used for LLM-as-a-judge evaluation. The model is forced to emit a strict JSON schema with five keys: `analysis`, `correctness`, `promptness`, `efficiency`, `overall` (latter four integers 1-5).

Action-Linked Classification Prompt (*gpt-5.5*, reasoning effort = high, temperature = 1.0)

```
You are classifying utterances from an AI assistant guiding a user through a physical procedure
(assembly, cooking, etc.) in an egocentric task-guidance setting.

For each utterance you receive:
- the utterance text
- a ±15s recipe context window (the user's nearby motor actions and other assistant utterances,
each with a timestamp).

Decide whether the utterance is a DIRECTIVE: an instruction that tells the user to start a physical
motor action they have not yet started.

INCLUDE as directive:
- Imperative: "Pull the lever", "Take out the battery"
- Permission-framed instruction: "Now you can change the battery"
- Conditional instruction: "If the screw is loose, tighten it"
- Action reminders: "Don't forget the cap"

EXCLUDE (not directive):
- Acknowledgement: "Okay", "Yes", "Great", "Thank you", "Mm-hm"
- Confirmation: "That's right", "Correct"
- Commentary / observation: "It's fine", "It looks loose"
- Pure deictic without action verb: "This one", "Yes, here"
- Questions: "Did you find it?"
- Greetings / closing remarks: "We are done", "Nice work"

Use "ambiguous" only when genuinely undecidable from text + context.

Respond as JSON exactly matching the schema. Keep rationale ≤ 30 words.
```

Figure 6: Prompt used for action-linked utterance classification with majority voting (strict-majority over 5 independent calls at temperature 1.0; ties resolved by the most frequent label). Response is enforced as strict JSON `{is_directive ∈ {yes, no, ambiguous}, rationale}`.

Utterance Type Classification Prompt (*gpt-5.6-luna*, 5-way majority voting)

```
You label single utterances spoken during a procedural physical task (assembling furniture, operating
a camera, making coffee...). The speaker guides a person who is performing the task and who is
wearing a head-mounted camera. The speaker may be a human instructor or an AI assistant; label the
utterance itself and do not speculate about who produced it.

Assign exactly one label.

GUIDANCE: the utterance tells the person what to do next, how to do it, what to avoid, or what to
fix. The person could act on it. This includes instructions, suggestions, warnings, corrections,
and reminders.
"Now attach the leg to the corner bracket."
"Careful, that screw goes in the other way."
"You will need the small hex key for this one."
"Try turning it counterclockwise instead."

NARRATION: the utterance describes the scene, an object, or what the person is doing, without asking
for any action. Typically third person or purely observational.
"The person is holding a screwdriver."
"There is a box of screws on the table."
"He is removing the lens cover."

OTHER: anything else. This includes acknowledgments and praise with no action attached ("Okay",
"Great job", "Mhm"), confirmations and answers ("Yes", "That one"), greetings and closings, meta-talk
about the conversation or recording, questions that request information rather than give it, filler,
and text that is unintelligible or in the wrong language.

Boundary rules.
1. Praise or acknowledgment alone is OTHER. It becomes GUIDANCE only when it is attached to an
instruction ("Good, now tighten it").
2. A description that implies an obvious next action is still NARRATION unless the action is actually
stated.
3. Judge only the text given. Do not infer intent from surrounding context you cannot see.

Return strict JSON: {"label": "GUIDANCE" | "NARRATION" | "OTHER", "reason": "≤ 15 words"}
```

Figure 7: Prompt used for utterance type classification. The label is the strict majority over 5 independent calls.

word with Qwen3-ForcedAligner-0.6B. Its duration is therefore the span over which the model is audible, which is shorter than the synthesized audio file since that also contains leading and trailing silence. The three quantities satisfy end-vs-action = start-vs-action - duration for every pair, and hence for their means, though not for their medians. The per-model decomposition is reported in Appendix A.6.2.

A.5 Human Evaluation

The full human evaluation results are described in Section 5.3. Each clip is centered on a ground truth instructor utterance and the user action it prompts, with temporal margins on both sides for context. We remove the human instructor's speech and overlay the spoken responses of two anonymous assistants, keeping the original user speech and background audio, so that the two versions of a clip differ only in the assistant.

We recruit 100 workers per comparison on Amazon Mechanical Turk. Each HIT contains three real A/B comparisons and one attention test (ground truth vs silence); workers who select the wrong answer on the test are excluded from the final scoring, leaving 252–267 valid judgments per comparison. A/B sides are randomized per question, and workers must achieve at least 50% cumulative playback on both clips before submitting. The total cost across all three comparisons is approximately $200. The evaluation template is shown in Figure 8.

A.6 Additional Results

A.6.1 Results on the `text-llm` Channel

Table 7 repeats the evaluation of Table 1 on `text-llm`, the text a model generates internally, rather than `text-asr`, the transcription of the speech it actually generates. Most content and judge metrics are slightly higher on `text-llm` for models that generate speech, as the text is not subject to TTS or ASR noise. End-vs-action also shifts between the two channels. The end of an utterance is measured from the generated audio in both cases, so it does not change. The onset does: on `text-asr` it is where the speech is first heard, and on `text-llm` it is the chunk where the model emits the text. Because predictions are matched to references by onset, the two channels score slightly different sets of interventions. The cascade row

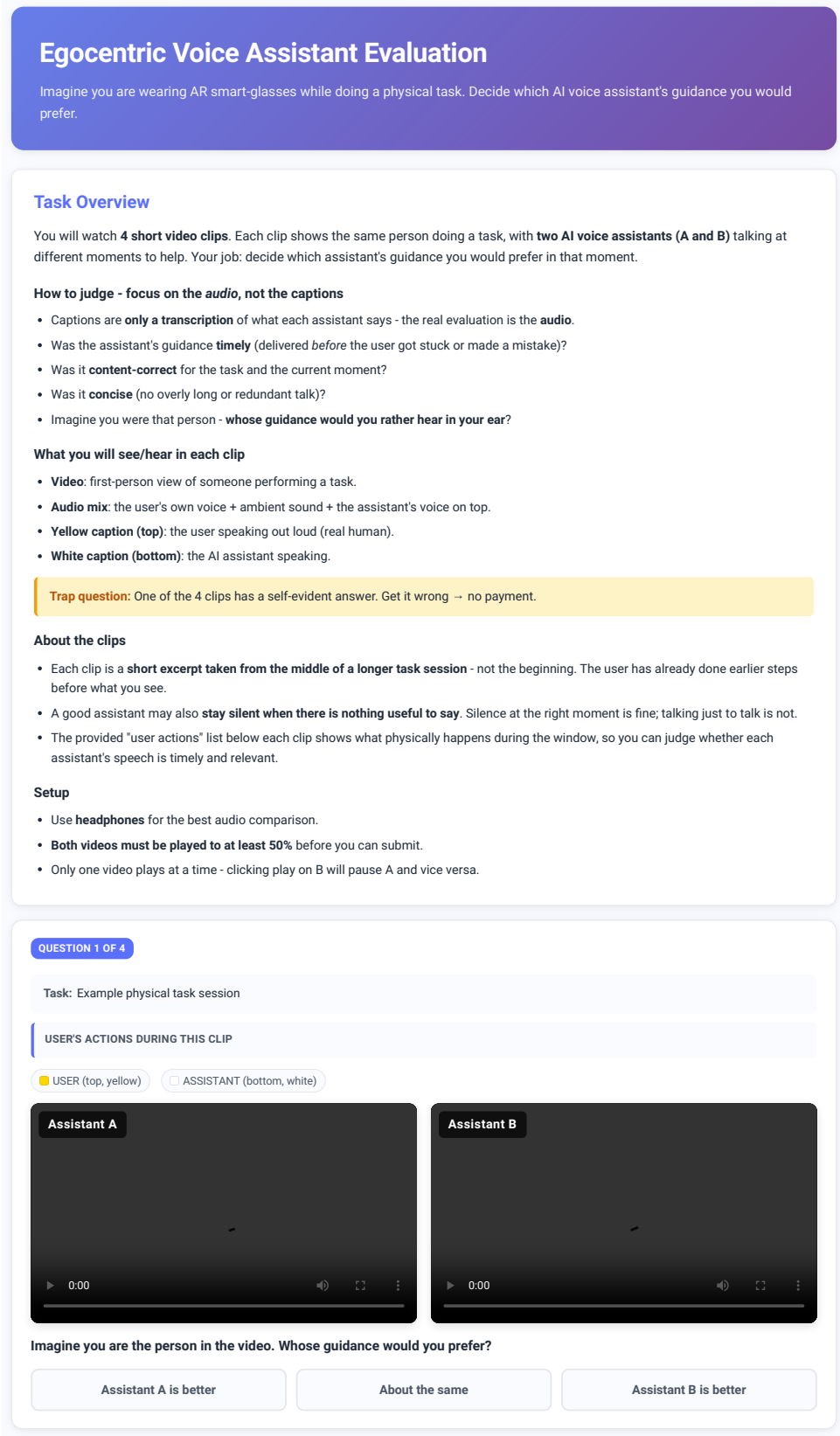


Figure 8: Human evaluation interface on Amazon Mechanical Turk. Evaluators watch two versions of the same egocentric clip with different assistant speech overlays and synchronized captions, then select which assistant (A or B) provides more helpful and timely guidance, or indicate a tie.

Model	Onset Timing			Content	LLM Judge (1–5)				End Timing
	Prec.	Rec.	F1	Cosine	Correct.	Prompt.	Effic.	Overall	end-vs-act (s)
Gemini Live	.364	.019	.022	.453	2.24	2.39	2.82	2.06	−3.65 / −3.62
`gpt-realtime`	.285	.024	.030	.499	2.24	2.70	2.54	2.20	−7.40 / −7.46
`gpt-realtime-2`	.256	.019	.024	.457	2.32	2.30	2.22	2.14	−16.0 / −16.8
ProAssist†	.480	.158	.195	.594	2.34	2.63	2.99	2.06	–
+ ASR/TTS cascade	.479	.158	.195	.585	2.35	2.63	2.97	2.07	−0.44 / −1.56
MiniCPM-o 4.5	.296	.111	.122	.538	1.76	1.73	1.99	1.58	−4.81 / −1.18
EgoVoice-SFT	.488	.241	.284	.707	2.59	2.74	2.77	2.37	+1.42 / +1.10
EgoVoice-DPO	.521	.302	.346	.674	2.61	2.68	2.71	2.39	+1.25 / +1.04

Table 7: Results on 170 benchmark sessions (`text-llm`). † indicates a text-only baseline. On this channel the cascade row is ProAssist reading ASR transcripts of the user's speech instead of ground-truth transcripts, so the two rows differ only in that input. Notation otherwise follows Table 1.

Model	n	Duration	Start-vs-action (s)	End-vs-action (s)
Gemini Live	3	6.64 / 6.48	+3.44 / +3.04	−3.20 / −2.26
`gpt-realtime`	30	11.73 / 10.84	+4.34 / +4.05	−7.40 / −7.46
`gpt-realtime-2`	22	20.39 / 20.20	+4.33 / +4.29	−16.1 / −16.8
ProAssist + ASR/TTS cascade	92	6.38 / 6.20	+5.94 / +4.82	−0.44 / −1.56
MiniCPM-o 4.5	99	9.87 / 6.00	+5.18 / +4.70	−4.69 / −0.40
EgoVoice-SFT	255	2.99 / 2.32	+4.55 / +4.38	+1.57 / +1.52
EgoVoice-DPO	287	2.93 / 2.56	+4.68 / +4.30	+1.75 / +1.62
Human instructor	942	3.05 / 2.60	+4.74 / +4.37	+1.69 / +1.38

Table 8: Decomposition of end-vs-action on the action-linked subset (T=5), in seconds, as mean / median. n is the number of matched interventions, which differs across models.

also serves as an ASR control here: it is ProAssist reading Qwen3-ASR-1.7B transcripts of the user's speech instead of ground-truth ones, and onset F1 is unchanged at 0.195 while cosine drops by only 0.009. The relative ranking is largely consistent between the two channels. Audio samples of our dataset, qualitative comparisons between EgoVoice and baselines, and additional examples are available on the project page.[9]

A.6.2 End-of-Utterance Decomposition

Table 8 separates end-vs-action into the two quantities that produce it. Every system starts speaking at a broadly similar point relative to the user's action, between +3.4 and +5.9 seconds. What differs is how long each one talks. EgoVoice-SFT and EgoVoice-DPO speak for 2.99 and 2.93 seconds on average, close to the instructor's 3.05, while the cascade takes 6.38 seconds, MiniCPM-o 4.5 takes 9.87, and `gpt-realtime-2` takes 20.39. Since end-vs-action is start-vs-action minus this duration, utterance length alone accounts for the sign of the result. Deciding when to speak is not the difficult part; finishing in time is.

[9] https://egocentricvoice.github.io/

Figure 9 shows the individual interventions behind these numbers. The instructor almost never runs late, and the two EgoVoice variants have the same shape with a heavier left tail. The baselines differ in kind rather than degree: the cascade and MiniCPM-o 4.5 sit astride zero, and the three closed-source APIs fall almost entirely below it. The figure also shows why we report medians alongside means. MiniCPM-o 4.5 has a mean of -4.69 but a median of only -0.40, because a handful of very late endings pull the average down.

A.6.3 Utterance Statistics

Table 9 reports how often each system speaks and how much it says. The two EgoVoice variants match the instructor's turn structure closely, at 8.1 and 7.8 words per utterance against 8.3, with a similar share of brief acknowledgements. They still speak about half as often as the instructor. Every baseline takes far longer turns, from 16.6 words for ProAssist to 51.6 for `gpt-realtime-2`, and almost never produces a short acknowledgement. This is the turn structure that the Efficiency axis of the LLM judge rewards, and it is also what pushes the baselines past the user's next action in Table 8.

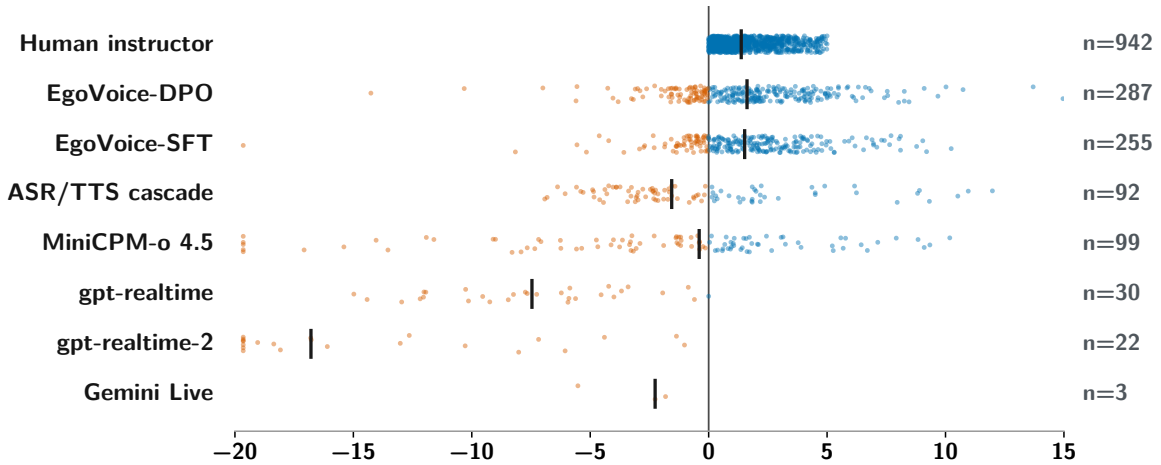


Figure 9: End-vs-action for every matched action-linked intervention (T=5, `text-asr`), in seconds. Each dot is one intervention; blue means the model finished speaking before the user began the linked action, orange means it was still speaking. The vertical tick marks the median and n is the number of matched interventions. Values outside the plotted range are clipped to it.

Model	Utt.	Words	≤3 words
Gemini Live	0.3	24.2	2.3%
`gpt-realtime`	0.9	37.7	0.0%
`gpt-realtime-2`	0.8	51.6	0.0%
ProAssist†	2.2	16.6	0.8%
+ ASR/TTS cascade	2.1	16.7	1.9%
MiniCPM-o 4.5	3.4	38.2	8.2%
EgoVoice-SFT	4.6	8.1	20.2%
EgoVoice-DPO	5.6	7.8	19.3%
Human instructor	11.3	8.3	17.9%

Table 9: Utterance statistics on the `text-llm` channel: utterances per session, mean words per utterance, and the share of utterances of three words or fewer.

Comparison	n	Diff	p
Onset F1			
DPO vs SFT	170	+0.050	0.0007
DPO vs MiniCPM	170	+0.211	6.0e−19
SFT vs MiniCPM	170	+0.161	1.2e−13
DPO vs ProAssist	170	+0.129	1.1e−08
SFT vs ProAssist	170	+0.079	7.9e−05
LLM Judge (Overall)			
DPO vs SFT	152	−0.072	0.42
DPO vs MiniCPM	156	+0.776	1.1e−13
SFT vs MiniCPM	149	+0.832	3.6e−13
DPO vs ProAssist	160	+0.094	0.38
SFT vs ProAssist	153	+0.157	0.12

Table 10: Paired Wilcoxon signed-rank tests over sessions. Onset F1 is defined for all 170 sessions, while judge scores exist only for sessions in which both models speak, so n varies. Comparisons against ProAssist use `text-llm` on both sides.

Counts here are on `text-llm`, so they are lower than the `text-asr` counts quoted in Appendix A.2 for MiniCPM-o 4.5.

A.6.4 Significance Tests

Table 10 collects the tests reported in Section 5. Onset F1 separates every pair of systems we compare. The LLM judge separates both EgoVoice variants from the zero-shot backbone, but neither from ProAssist nor from each other. We do not test cosine similarity. Unlike the other two metrics, it is defined per matched pair, and each model is scored on a different set of pairs, so there is no common set on which to run a paired test.

A.6.5 Qualitative Examples and Failure Modes

Figures 10 to 12 show six EgoVoice-DPO interventions in the format of Figure 3, with two additions: the instructor's own speech is drawn at the bottom of each panel, so the two waveforms can be compared directly, and the start of the user's linked action is marked. Figure 10 shows the intended behavior, where the model speaks at the same moment as the instructor, delivers the same instruction, and stops before the user acts.

The two failure modes we measure follow. In Figure 11 the onset and the content are both right, but the model speaks for 5.8 and 4.0 seconds where the instructor took 1.6 and 1.4, so the user is already acting while the assistant is still talking. This is the failure behind the 28.2% late-ending rate. In Figure 12 the model guides the wrong step, delivering an instruction that belongs 14 and 17 seconds earlier in the task. These are the pairs for which a

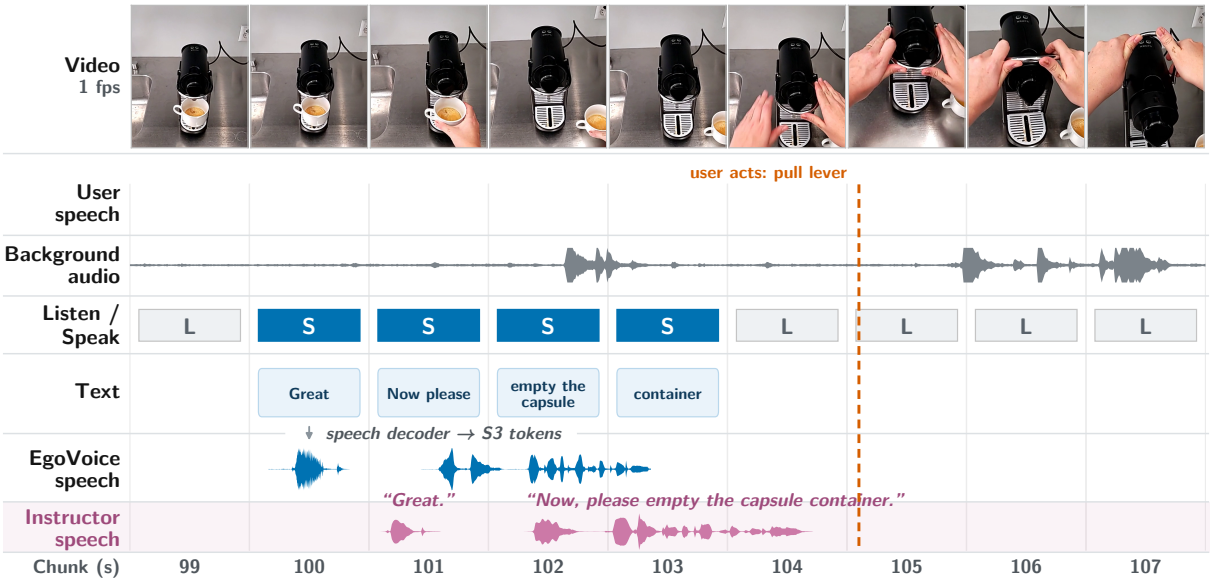


Figure 10: Two interventions that succeed. Each panel follows Figure 3, with the instructor's own speech added at the bottom (shaded) and the start of the user's linked action marked. EgoVoice-DPO speaks at the same moment as the instructor, says the same thing, and finishes before the user acts. The system prompt lists the task steps in third person, so the model has to choose which step is due, phrase it as an instruction, and time it.

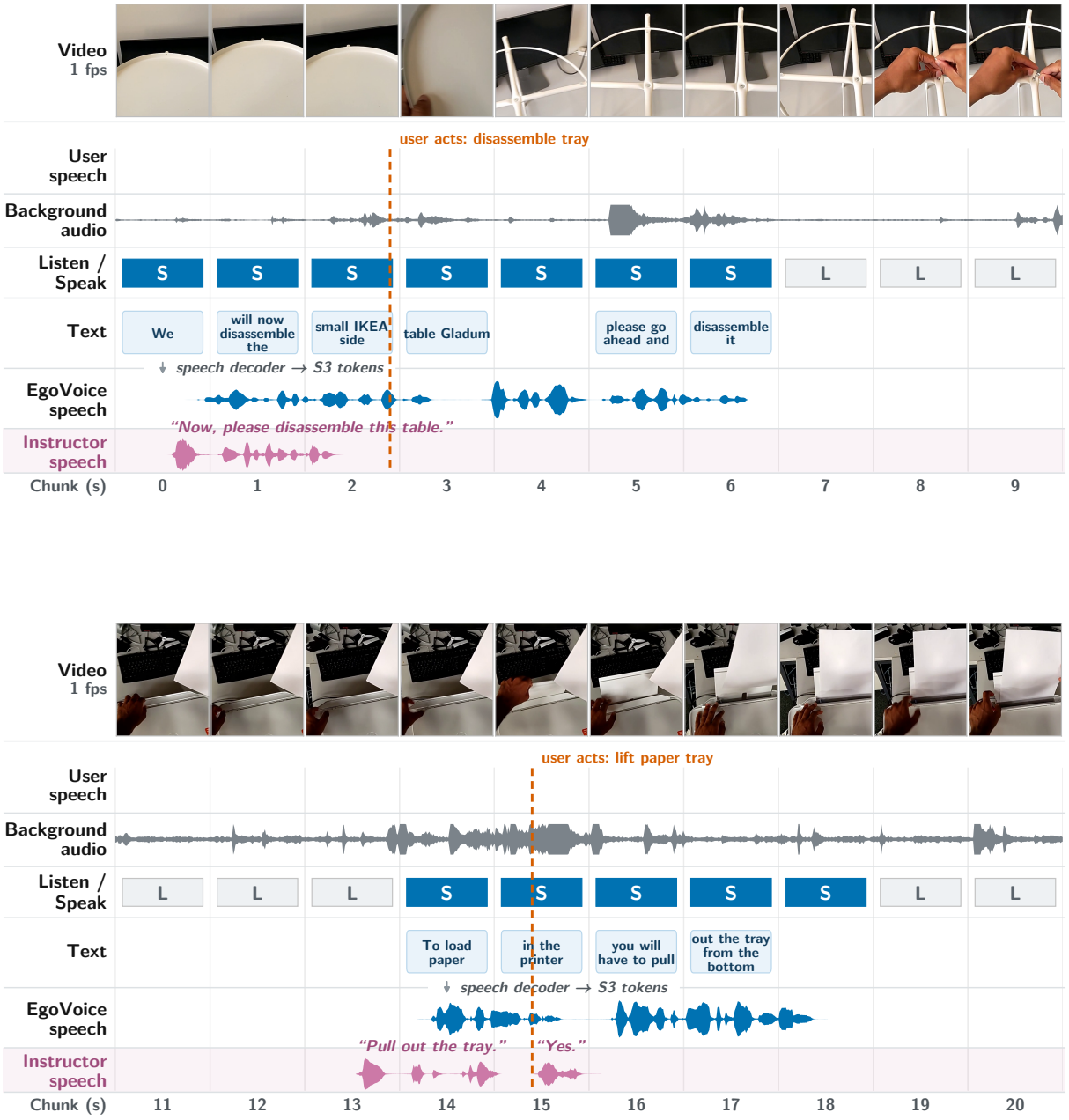
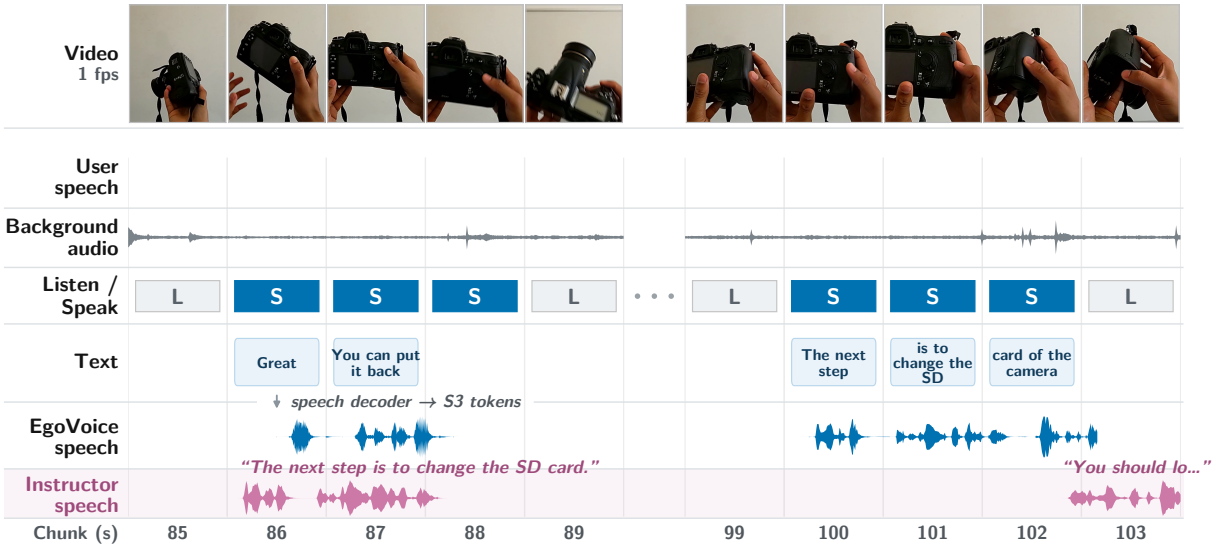


Figure 11: Two interventions that end too late. The onset and the content are right, but the model's speech runs well past the instructor's and is still playing when the user begins the action.



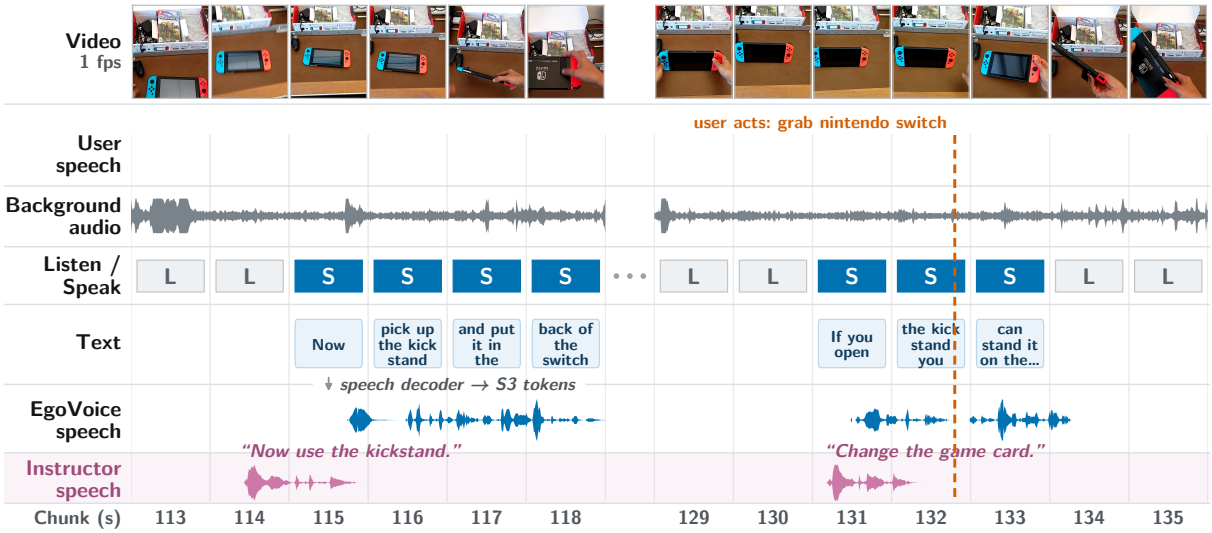


Figure 12: Two interventions that guide the wrong step. The model delivers an instruction that belongs elsewhere in the task: in the first, one the instructor gave 14 seconds earlier; in the second, one the model itself gave 17 seconds earlier. In both cases the instructor has already moved on.

Artifact	License	URL
Datasets		
HoloAssist	CDLAv2	https://holoassist.github.io
Multilingual LibriSpeech	CC BY 4.0	https://www.openslr.org/94
Backbone & Training		
MiniCPM-o 4.5	apache-2.0	https://huggingface.co/openbmb/MiniCPM-o-4_5
Data Construction		
IndexTTS2	Bilibili Model License	https://huggingface.co/IndexTeam/IndexTTS-2
SAM-Audio (large-tv)	SAM License	https://huggingface.co/facebook/sam-audio-large-tv
Silero VAD	MIT	https://github.com/snakers4/silero-vad
Evaluation		
Qwen3-ASR-1.7B	apache-2.0	https://huggingface.co/Qwen/Qwen3-ASR-1.7B
Qwen3-ForcedAligner-0.6B	apache-2.0	https://huggingface.co/Qwen/Qwen3-ForcedAligner-0.6B
Qwen3-Embedding-0.6B	apache-2.0	https://huggingface.co/Qwen/Qwen3-Embedding-0.6B
gpt-5.4-mini (judge)	Proprietary	https://developers.openai.com/api/docs/models/gpt-5.4-mini
gpt-5.5 (action-linked classifier)	Proprietary	https://developers.openai.com/api/docs/models/gpt-5.5
gpt-5.6-luna (utterance type classifier)	Proprietary	https://developers.openai.com/api/docs/models/gpt-5.6-luna
Baselines		
Gemini Live API	Proprietary	https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash-live-api
`gpt-realtime`	Proprietary	https://developers.openai.com/api/docs/models/gpt-realtime
`gpt-realtime-2`	Proprietary	https://developers.openai.com/api/docs/models/gpt-realtime-2
ProAssist	Not specified	https://huggingface.co/594zyc/ProAssist-Model-L4096-I10

Table 11: All artifacts used in this work with their licenses and sources.

neighbouring reference is closer than the matched one.

A third failure comes from the backbone rather than from intervention behavior. In 47 of 1,090 utterances the model does not stop, producing degenerate output such as *"Or this, this, put it on the table. Put it on the table. Put it on"* over 21 seconds. Language switching appears in 13 utterances, including a 373-second Chinese repetition loop. These artifacts are inherited from MiniCPM-o 4.5 and are discussed in the Limitations section.

A.7 Artifacts and Licenses

Table 11 lists all datasets, models, and tools used in this work with their licenses. We use all artifacts for research purposes consistent with their intended use. Our dataset is derived from HoloAssist, which is released under CDLAv2, a permissive license that allows redistribution and derivative works. The original HoloAssist recordings anonymize speaker identities through pitch shifting. Our speech resynthesis replaces all original voices with zero-shot TTS output using unrelated reference speakers from Multilingual LibriSpeech, so no original speaker identity is recoverable from our released data.

A.8 AI Assistant Usage

We use *gpt-5.4-mini* for transcript normalization (Section 3.1) and LLM-as-a-judge evaluation (Appendix A.4), *gpt-5.5* for action-linked utterance classification, and *gpt-5.6-luna* for utterance type classification (Appendix A.4). We additionally use an AI assistant for proofreading the manuscript and the human evaluation survey template, for drafting analysis and plotting code, and for generating the initial draft of Figure 1.