HEIR: HARNESS EGOCENTRIC INTENT FOR HUMAN-ROBOT INTERACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

For robots to assist, serve, and collaborate with humans, they must not only interpret human requests conveyed through language and multimodal intentional cues, but also execute long-horizon tasks that interleave navigation and manipulation. In this paper, we introduce **HEIR**, a resource for studying how to **H**arness **E**gocentric **I**ntent for Human–**R**obot Interaction. HEIR pairs human speech, measured gaze, and egocentric video with robot observations and teleoperated navigation&manipulation trajectories, while each session preserves successive episodes, with each request fulfilled through multiple execution subtasks. The captured data, together with human-authored annotations, support the study of intent grounding, execution planning, and task completion within a unified interaction setting. To demonstrate the value of HEIR, we conduct extensive offline experiments to evaluate how models benefit from additional human-centric supervision across these tasks, together with real-robot experiments that assess execution task completion. HEIR thus provides a valuable resource for studying user-centric robot interaction in naturalistic household environments. Our anonymous demo and dataset are available online.

1 INTRODUCTION

The research community has witnessed revolutionary progress driven by data-centric approaches, enabling the rapid advancement of large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2023; Grattafiori et al., 2024; Rae et al., 2021; Hoffmann et al., 2022)and vision-language models (VLMs) (Radford et al., 2021; Alayrac et al., 2022; Li et al., 2023b). Tremendous efforts have been invested in reproducing this success in embodied AI, leading to rapid advances in vision-language-action models (VLAs) (Brohan et al., 2023; Kim et al., 2024; Team et al., 2024; Black et al., 2024; Bjorck et al., 2025) and world-action models (WAMs) (Ye et al., 2026; Li et al., 2026c; Yuan et al., 2026). More recently, inspired by LLM agents that harness external tools, agentic robot systems (Ahn et al., 2022; Huang et al., 2022) have been developed to orchestrate multiple execution modules for long-horizon tasks.

Much of this progress remains fundamentally task-centric, where robots execute given instructions while humans remain peripheral to the formulation. This paradigm is insufficient for robots intended to serve, assist, and collaborate with people in real-world scenarios, where humans are not merely obstacles to avoid, but active participants whose behaviors, intentions, and requests continuously shape the task. We therefore argue that *embodied AI should increasingly embrace a user-centric perspective, placing situated human context at the core of robot decision-making and execution*.

Studying this new paradigm in the data-driven era requires datasets that capture not only robot sensory and action signals, but also how humans perceive, communicate, and interact with robots throughout task execution. Existing Human–Robot Interaction (HRI) datasets (Scalise et al., 2018; Yasar et al., 2024), however, are largely collected in controlled tabletop settings and focus on short-horizon interactions, providing limited coverage of situated human behaviors. Meanwhile, egocentric vision has been increasingly explored for robot learning (Wang et al., 2023; Grauman et al., 2022), yet limited effort has studied how intentional cues from the human perspective can complement robot observations during HRI.

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

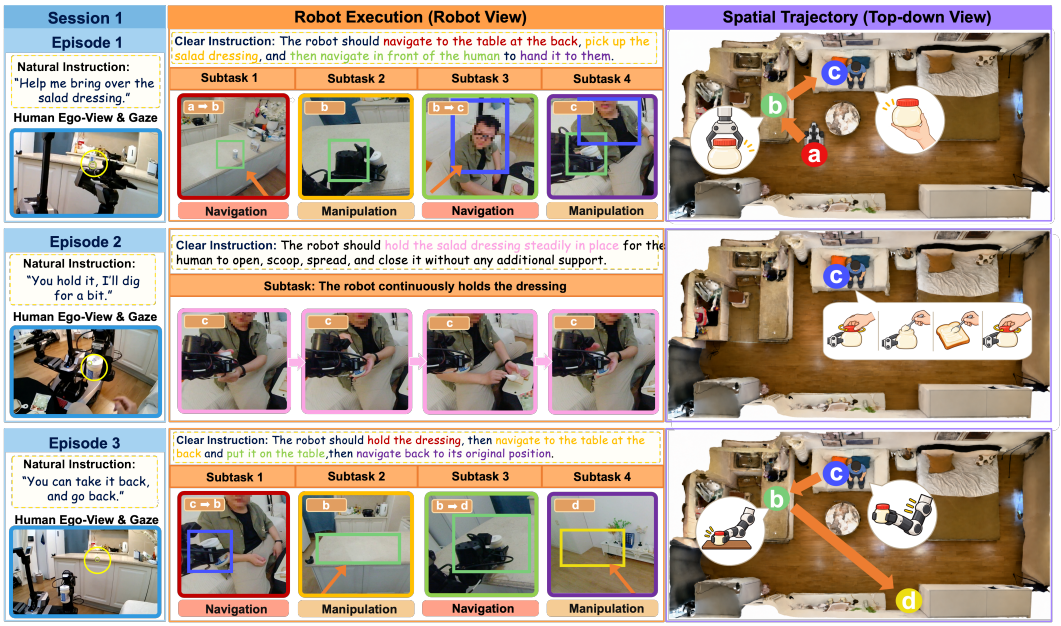

Figure 1: *Example long-horizon HRI session in HEIR*. Eech one of our recorded session contains successive episodes, each initiated by a natural human request grounded in the user's egocentric view and gaze. Each request is mapped to a clarified task specification and fulfilled through one or more navigation and manipulation subtasks, while the top-down trajectory visualizes the corresponding spatial execution and state transitions across episodes.

To bridge this gap, we introduce HEIR, **H**arnessing **E**gocentric **I**ntent for Human–**R**obot Interactions, a dataset for studying user-centric human–robot interaction during long-horizon tasks in naturalistic household environments. As illustrated in Fig. 1, HEIR pairs human egocentric video, measured gaze, and speech with robot observations and teleoperated navigation–manipulation trajectories. These recordings capture the human and robot context surrounding each request, including referents visible only from the human perspective, while each continuous session preserves both multi-stage execution and state changes across requests. Our dataset further provides annotations of task intent, target objects, destinations, nominal plans, and realized navigation–manipulation stages, thereby linking situated human context to task semantics and robot behavior.

HEIR supports a hierarchical set of tasks spanning VLM-based user intent inference, subtask planning, and navigation and manipulation execution. In this paper, we evaluate how additional user-centric cues benefit intent modeling and long-horizon planning for HRI, benchmark prevailing navigation and manipulation methods through offline evaluation, and further report task success rates in real-robot experiments. We also provide episode-level experimental results and video demonstrations on long-horizon tasks to demonstrate how HEIR establishes an empirical foundation for studying user-centric robot interaction.

2 Related Work

Large-scale datasets have rapidly advanced robot execution (Anderson et al., 2018; Ku et al., 2020; Fang et al., 2024; Walke et al., 2023). However, resources for studying natural HRI remain scarce. HABIT structures human–robot co-execution around collaborator, coworker, and supervisor roles, supporting the study of synchronization, yielding, and gesture grounding (Song et al., 2026). HARMONIC records gaze, human control inputs, and robot state during assistive feeding; NatSGLD pairs spoken and gestural commands with Linear Temporal Logic interpretations and simulated robot demonstrations (Newman et al., 2022; Shrestha et al., 2025). HandMeThat grounds ambiguous requests in human action history, FISER separates intent inference from planning on HandMeThat, and REI-Bench varies referential vagueness and dialogue context (Wan et al., 2022; 2025; Jiang et al., 2026). WatchAct links real human-action videos to executable manipulation tasks and separately evaluates video-to-plan reasoning, execution under oracle plans, and integrated pipelines (Li et al., 2026a). Table 1 compares HEIR with representative robot datasets and systems.

Table 1: *Comparison of representative robot datasets and systems.* We compare coverage of real-world data, paired human and robot (H&R) views, gaze, speech, intent annotations, long-horizon interaction, and support for both navigation and manipulation execution.

Work	Real-world	H&R views	Gaze	Speech	Intent annot.	Long horizon	Navi.& Mani.	# of subtasks
Gaze2Act (Zuo et al., 2026)	✓	✓	✓	✓	✓	×	×	653
HABIT (Song et al., 2026)	✓	✓	×	×	✓	✓	×	10,563
HARMONIC (Newman et al., 2022)	✓	×	✓	×	×	×	×	480
NatSGLD (Shrestha et al., 2025)	×	✓	×	✓	✓	×	×	1143
RoboVQA (Sermanet et al., 2024)	✓	×	×	×	✓	✓	✓	5246
RoboCasa365 (Nasiriany et al., 2026)	×	×	×	×	×	✓	✓	55000
WatchAct (Li et al., 2026a)	×	×	×	×	✓	✓	×	3000
TEACh (Padmakumar et al., 2022)	×	×	×	×	✓	✓	✓	3,047
EDITH (Lee et al., 2026)	✓	✓	✓	✓	✓	×	×	280
Ours	✓	✓	✓	✓	✓	✓	✓	1836

Notably, HEIR stands out as the only dataset that jointly combines paired human–robot views, gaze, speech, intent annotations, long-horizon interaction, and both manipulation and navigation in real-world settings.

Our work is also closely related to human intent grounding for HRI and long-horizon robot execution; we provide additional discussion of these directions in the Appendix A.

3 Data Collection and Annotation Protocol

Our **HEIR** is designed to study how robots can perform human–robot interaction in household environments by responding to situated, natural human requests. We provide an overview of HEIR in Fig. 2. It jointly captures the user's egocentric view with eye tracking, remotely controlled robot navigation and manipulation trajectories together with onboard sensory signals, and 3D scans of the staged environment.

3.1 Task Formulation and Interaction Structure

Interaction hierarchy. HEIR organizes continuous household interactions into *sessions*, each containing an ordered sequence of request-level *episodes*. Each episode associates one spoken request with its subsequent robot execution, segmented into navigation and manipulation *subtasks*. A subtask spans a continuous execution segment; a manipulation subtask can involve multiple objects or both arms. The number and order of subtasks vary with the request and the robot's state. Episodes within a session share an evolving physical environment, allowing later requests to refer to objects or locations involved in earlier interactions.

Human-Robot Interaction records. Within a session $\mathcal{S} = (\mathcal{E}_1, \ldots, \mathcal{E}_K)$, the captured data for the k-th episode, $k \in \{1, \ldots, K\}$, are represented as a triplet:

$$\mathcal{E}_k = (X_k^H, X_k^R), \qquad X_k^H = (V_k^H, A_k^H, G_k^H). \tag{1}$$

X_k^H represents the human-centric context, comprising the user's egocentric video V_k^H, recorded speech audio A_k^H, and temporally aligned gaze measurements G_k^H. X_k^R comprises robot-view video and state–action streams. These temporally aligned records cover the request context and subsequent execution, preserving human-side visual context and interaction cues alongside robot behavior.

Annotation. The annotation record Y_k links each request to its intended task and each execution segment to a subtask objective. Intent annotations specify the grounded task, manipulated objects, goal entities, and nominal navigation–manipulation plan, while execution annotations provide realized subtask types, temporal boundaries, subgoals, and completion conditions. The nominal plan captures the intended task structure, whereas the realized sequence reflects execution-specific adjustments. Details are provided in Section 3.3.

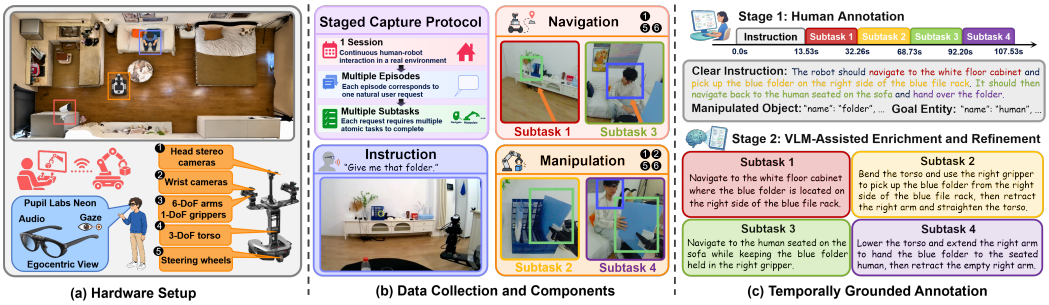

Figure 2: *Overview of the HEIR data components*. (a) Hardware setup combining human egocentric sensing with gaze and audio, together with robot-side sensory signals and teleoperated execution. (b) Each continuous session contains multiple episodes initiated by natural human requests, with each episode composed of navigation and manipulation subtasks. (c)Human annotators first create temporally grounded task annotations from the recorded interactions, which are then refined by an LLM-assisted refinement model into structured instructions and subtask descriptions.

3.2 Multimodal Data Collection

Task Scenarios. We collect continuous household interactions using predefined scenarios spanning daily living, kitchen activities, and home-office assistance. These scenarios organize object retrieval, handover, relocation, and cooperative manipulation into sequences of requests. For example, a cleaning scenario involves fetching supplies, handing them to the user, and subsequently disposing of waste or returning tools. Across recordings of the same scenario, the core task goals remain consistent, while object positions vary within the same general areas.

Hardware Setup. We collect participant-side recordings using Pupil Labs Neon eye-tracking glasses and adopt a Galaxea R1 Lite (2025 version) as mobile dual-arm robot platform. The human-side recordings contain egocentric video, speech, and measured gaze. Robot-side recordings include four video streams from two top cameras and two wrist cameras, arm and gripper actions, and state measurements for the arms, grippers, torso, and mobile base.

Stage Protocol. Participants express requests in their own words while attending to task-relevant objects and locations. A separate operator with a global scene view teleoperates the robot to execute navigation and manipulation in response to each request and subsequent interaction cues. Notably, even for the same retrieval goal, interactions may unfold differently: participants may verbally request individual handovers or indicate preferences through reaching gestures. Robot-view and human-egocentric videos capture these behaviors and resulting handovers, preserving the interaction context underlying each demonstrated response.

Temporal Alignment. Neon scene video is recorded at a nominal 30 FPS. Stopwatch readings shared by both cameras at the start and end define a linear mapping to the robot's 30 FPS timeline, correcting initial offsets and approximately 4 s of accumulated drift over 10 min (0.67%). We resample the human video and time-stretch its audio while preserving pitch, with gaze samples matched to human frames by timestamp.

3.3 Annotation Procedure

Human annotation. As illustrated in Fig. 2, annotators ground each request using speech, measured gaze, and visual context, labeling its evidence intervals and structured intent, and whether the grounded human-intent target is visible to the robot at the time of instruction. Annotators also label the types and temporal boundaries of navigation and manipulation subtasks. Robot video provides evidence of interaction objects and task outcomes, complementing the motion and configuration captured by state–action records.

VLM-assisted enrichment and refinement. To support language-conditioned execution and subtask planning, we augment each execution segment with a *subgoal* specifying what to accomplish and a *completion condition* specifying when it is complete. We use GPT-5.6 to generate these annotations from human annotation, robot-view video, and aligned state–action records, thereby linking visual evidence of object interactions to robot movements. Manipulation labels describe the demonstrated actions, manipulated objects, executing arms or grippers, and relevant torso ad-

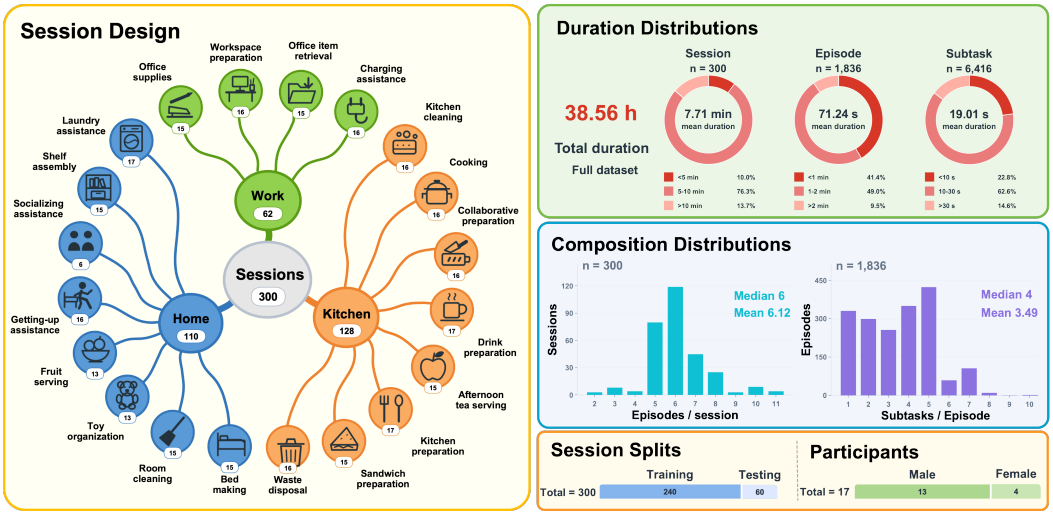

Figure 3: **Statistics and composition of HEIR**. Our dataset contains 300 sessions across 16 task categories, totaling 38.56 hours, 1,836 episodes, and 6,416 subtasks. We further report interaction composition, train/test splits, and participant statistics.

justments; navigation labels specify destinations and carried-object constraints. Subtask boundaries and execution order are preserved. The VLM also corrects grammatical and typographical errors in human-authored intent descriptions without changing the original meaning.

Quality assurance. We randomly inspect 20% of request-level episodes, checking intent annotations and the subgoals and completion conditions of all their subtasks against the recordings. Flagged or ambiguous records receive additional review. Annotation conventions and review procedures are detailed in Appendix B.5.

3.4 Navigation Geometry and External Validation

Trajectory Reconstruction and Validation. We fuse MapAnything (Keetha et al., 2026) visual estimates with timestamped wheel odometry recorded during teleoperation to reconstruct planar base trajectories, aligned frame-wise with robot-view video and navigation annotations. The reconstructed scene geometry is registered to MID-360 scans. Validation uses HTC VIVE Tracker measurements from several recordings. A 16,323-frame reference recording yields 8.55 cm planar position RMSE with a recording-fitted Tracker-to-base transform. Across 76 one-second windows (Tracker net translation $\geq$5 cm), local endpoint error averages 1.63 cm (95th percentile: 3.50 cm). Separately, the filtered Tracker measurement-point and fused base trajectories yield a median XY displacement-direction cosine similarity of 0.9992 after planar rigid alignment without scale fitting, over 105 one-second windows with net displacement $\geq$5 cm in both trajectories. These results support using the reconstructed trajectories as motion references for navigation demonstrations in the recorded environment.(Fig. 4).

3.5 Dataset Statistics

As shown in Fig. 3, HEIR is collected from 17 participants and comprises 300 interaction sessions spanning 20 everyday task categories across home, work, and kitchen scenarios. In total, the dataset contains 38.56 hours of interaction, 1,836 episodes, and 6,416 execution subtasks, with each session averaging 7.71 minutes. Each session contains an average of 6.12 episodes, while each episode comprises 3.49 subtasks on average, reflecting the hierarchical and long-horizon nature of HEIR. For our experiments, we split HEIR at the session level into 240 training and 60 testing sessions.

4 TASK SETUP AND EVALUATION FRAMEWORK

4.1 HUMAN INTENT UNDERSTANDING

Task Formulation. For episode $\mathcal{E}_k$, given a spoken request, temporally ordered human-egocentric frames with measured gaze, corresponding robot observations from X_k^R, and the intent predictions from preceding episodes in the same session, the task is to infer the user's structured intent before execution. The output includes a grounded task description, manipulated objects, goal entities, and a nominal navigation–manipulation plan. The predicted intent is then used as task context for downstream online planning.

Metrics. We evaluate intent understanding using Slot micro-F1 with task-specific matching for structured extraction, following the slot-based evaluation paradigm of (Rastogi et al., 2020), Location Accuracy under our spatial-compatibility rules, our joint entity–visibility F1 for object identification and visibility, raw and torso-guarded Plan EM for exact phase-sequence agreement before and after state-based correction, and BERTScore F1 for semantic similarity between predicted and reference intent descriptions (Zhang et al., 2019).

4.2 SUBTASK PLANNING

Subtask Planning. Given the inferred episode-level intent, current robot observations and state, and execution history, the task is to determine the active navigation or manipulation subtask and its progression during closed-loop execution. At each decision point, the planner outputs one of three actions: *Continue*, which maintains the current subtask; *Switch*, which advances to the next subtask when the current one is complete but the overall request remains unfinished, and predicts its subgoal and completion condition; or *Done*, which terminates execution once the episode goal is satisfied. The subgoal specifies the immediate execution objective, while the completion condition defines the observable state used to assess progress. This formulation separates the nominal episode-level plan from state-dependent execution, allowing the planner to adapt subtask transitions to robot configuration and local reachability.

Metrics. We report *Continue*/*Switch*/*Done* accuracy (Acc) and unweighted class-averaged F1 (Macro-F1) (Opitz, 2024). On all ground-truth *Switch* cases, we measure next-subtask type accuracy (Next Type) and use an LLM judge (Zheng et al., 2023) to assess core subgoal correctness (actions, objects, and destinations; Core) and terminal-state predicate F1 (Completion F1), inspired by (Li et al., 2024). Incorrect decisions or subtask types score zero on these three metrics; Core and Completion F1 use the midpoints of unresolved-judgment bounds.

4.3 NAVIGATION

Task Formulation. Given a navigation subgoal, the task is to determine how the robot should move through the environment based on its current observations and state. A vision-language navigation (VLN) model interprets the subgoal together with visual and spatial context and outputs a sequence of control commands that guides the robot toward the target location while adapting to the observed environment. A Mid-360 LiDAR is additionally used for collision avoidance during navigation.

Metrics. We evaluate offline action prediction using single-step action accuracy (Chen et al., 2021) and macro-F1 over four action classes (*Forward*, *Left*, *Right*, and *Stop*). Termination is evaluated using *Stop* recall and F1 against terminal stationary intervals derived from recorded base poses and wheel-speed feedback within annotated navigation subtasks.

4.4 MANIPULATION

Task Formulation. Given a manipulation subgoal, the task is to generate robot manipulation actions using the current robot observations and state. This task can be addressed by either a vision-language-action (VLA) model or a world action model (WAM), where the output is a sequence of low-level control commands that specifies the robot end-effector motion and gripper actions required to accomplish the subgoal.

Table 2: *Input ablations for Gemma-4-31B.* ↑ *denotes the higher the better.* **Bold** indicates the best score across input conditions for each model. RV is N/A without robot views.

	Grounding		Planning	Visibility	Text
Input condition	Slot F1 ↑	Loc. Acc. ↑	Plan EM ↑	RV F1 ↑	BERTScore F1 ↑
Textual Instruction only	60.38	19.42	33.24	N/A	92.96
Robot View only	71.39	55.38	64.32	61.63	93.19
Human + Robot	75.74	72.18	73.51	60.94	**93.27**
Human + Robot + center prior	75.24	70.08	78.92	**64.10**	93.23
Human + Robot + gaze (full)	**77.80**	**77.95**	**79.19**	63.95	93.23

Metrics. We evaluate the first predicted action and 24-step chunks against recorded targets using joint MSE (rad^2) (James et al., 2021), FK-based end-effector position error (cm), and minimum relative rotation angle (°) in each arm-base frame (Lynch & Park, 2017). Errors are averaged over 12 arm joints or both end effectors, respectively, and over test samples. Chunk metrics additionally average over 24 steps, with position error following ADE (Gupta et al., 2018).

4.5 Real-Robot Evaluation

Task Formulation. For real-robot evaluation, we examine whether the robot can successfully fulfill the participant's natural request in each episode. At deployment time, such an end-to-end system must not only address all four aforementioned tasks, but also seamlessly integrate them within a hierarchical execution pipeline. We provide a system overview and algorithmic design in the Appendix C.1. Specifically, the intent reasoner first integrates human-centric context, robot observations, and interaction history to infer the grounded task and nominal plan. A closed-loop planner then updates execution from the current robot state, deciding whether to **Continue**, **Switch** subtasks, or **Done**. Navigation and manipulation subgoals are subsequently dispatched to the VLN and VLA/WAM backends, respectively.

Metrics. We report both task-level and episode-level success rates, and provide qualitative video demonstrations in the supplementary material.

5 Experiments and Analysis

In this section, we briefly summarize the key experimental insights and findings across the four long-horizon HRI tasks, while implementation details and complete experimental results are provided in the Appendix C and E, respectively.

5.1 Human Intent Understanding

Here, we use Gemma-4-31B (Team et al., 2026) as the base model, with additional experiments on Qwen3.8-27B (Qwen Team, 2026b) and InternVL3.5-38B (Wang et al., 2025) reported in the Appendix E. At a similar model scale, Gemma-4 slightly outperforms Qwen3.8 and substantially outperforms InternVL3.5 on human intent understanding. As these large pretrained models already achieve strong performance without task-specific training, we focus our analysis on the contributions of human-centric intentional cues and interaction history.

The role of Human-Centric Context. For efficiency, we omit interaction history from this evaluation. As shown in Tab. 2, human-centric intentional context substantially benefits intent parsing. The instruction-only baseline achieves the lowest accuracy, indicating that intent parsing is non-trivial and cannot be directly solved from language priors alone. Adding either the robot view or the human egocentric view significantly improves performance. And overlaying gaze on the egocentric frames provides further gains, particularly compared with using a center prior as a proxy for attention. Notably, combining all modalities yields the best overall performance.

The role of Intent History. We adopt the full-context setup to examine the role of intent history. Ground-truth history provides an upper bound on the benefit of perfect interaction memory, while

Table 3: *Intent understanding results for Gemma-4-31B under different history conditions.* ↑ denotes the higher the better. **Bold** indicates the best score without considering ground truth.

	Grounding		Planning	Visibility	Text
History	Slot F1 ↑	Loc. Acc. ↑	Plan EM ↑	RV F1 ↑	BERTScore F1 ↑
Ground-truth history	82.19	82.41	84.05	64.73	93.57
No history	77.80	77.95	79.19	**63.95**	93.23
Predicted history	**78.62**	**80.05**	**81.62**	63.67	**93.32**

Table 4: *Planner results with instruction-only and instruction-plus-intent inputs.* All models are fine-tuned on the HEIR training split. **Bold** indicates the higher score between the two input conditions within each model.

		Planning				Completion
Model	Input	Acc ↑	Macro-F1 ↑	Next Type ↑	Core ↑	Completion F1 ↑
Cosmos3-Nano	Instruction	64.90	65.51	77.59	48.56	45.18
(NVIDIA, 2026)	Instruction + intent	**65.62**	**66.19**	**82.18**	**64.37**	**52.24**
Qwen3.5-4B	Instruction	**69.71**	**71.93**	**75.29**	50.86	44.15
(Qwen Team, 2026a)	Instruction + intent	69.23	71.40	73.56	**59.48**	**46.84**
Gemma-4-E4B-it	Instruction	61.06	59.27	90.80	56.32	44.99
(Team et al., 2026)	Instruction + intent	**62.02**	**60.14**	**93.10**	**65.80**	**50.24**

regressively feeding predictions from preceding episodes also yields notable gains. These results suggest strong dependencies across naturally evolving interactions and validate the importance of modeling long-horizon interaction history in HEIR.

5.2 Subtask Planning

We assess the impact of explicit intent parsing on subtask planning by comparing fine-tuned planners with and without structured intent input. However, the gains are relatively modest, which we speculate is due to the limited long-context understanding capability of the smaller planner models. As shown in Tab. 4, adding intent generally improves planning quality, with consistent gains in core planning and completion prediction across all three models.

5.3 Robot Navigation

Although existing VLN models (Zhang et al., 2024; Wei et al., 2025; Team, 2025) exhibit notable generalization ability, their original checkpoints still degrade substantially in our setting. Fine-tuning on HEIR markedly improves both action prediction and stopping performance across all models, demonstrating the value of adapting VLN policies to naturalistic long-horizon HRI scenarios.

5.4 Robot Manipulation

We report robot manipulation performance for several prevailing VLA/WAM models after fine-tuning on HEIR, including DreamZero (Ye et al., 2026), π_0-FAST (Pertsch et al., 2025), π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), FastWAM (Yuan et al., 2026), and Lingbot-VLA2 (Wu et al., 2026). We found that existing open-source checkpoints exhibit limited zero-shot generalization to our naturalistic household scenarios, often leading to severe execution failures and making direct evaluation uninformative. We therefore focus on fine-tuned model performance. As shown in Tab. 6, stronger recent models consistently reduce both joint-space and end-effector prediction errors, with Lingbot-VLA2 achieving the best overall performance across first-step and action-chunk prediction metrics.

In our real-robot experiments, Lingbot-VLA2 also achieves the highest task-level success rate, showing a consistent trend between offline action prediction and online execution. This suggests that our

Table 5: *Offline VLN action prediction before and after fine-tuning with English subgoal instructions.* All scores are percentages; ↑ denotes higher is better. **Bold** denotes the better weight setting within each model block. Original denotes the released model without project LoRA; fine-tuned checkpoints are selected on validation. † denotes annotation-bounded STOP proxy scores.

		Action prediction		Stopping	
Model	Weights	Accuracy ↑	Macro-F1 ↑	STOP Recall† ↑	STOP F1† ↑
NaVid	Original	48.74	45.38	53.49	30.95
	Fine-tuned	**86.59**	**86.15**	**80.72**	**60.05**
StreamVLN	Original	46.84	39.65	19.39	20.34
	Fine-tuned	**86.36**	**86.20**	**75.49**	**67.02**
InternVLA-N1	Original	49.08	45.41	58.28	31.48
	Fine-tuned	**90.69**	**91.00**	**84.53**	**76.98**

Table 6: *Offline VLA/WAM action prediction across model variants.* All models are fine-tuned using ground-truth subgoal descriptions as instructions. All values are percentages (higher is better).

	First output			Action chunk		
Model	Joint MSE ↓ (rad^2)	EEF Pos. ↓ (cm)	EEF Rot. ↓ (°)	Joint MSE ↓ (rad^2)	EEF Pos. ↓ (cm)	EEF Rot. ↓ (°)
DreamZero	0.002809	1.489	3.455	0.013215	2.604	5.244
π_0-FAST	0.001644	0.808	1.093	0.009090	1.466	1.858
π_0	0.000588	0.396	0.653	0.003688	0.937	1.499
$\pi_{0.5}$	0.000468	0.373	0.652	0.003398	0.911	1.519
FastWAM	0.000374	0.369	0.673	0.002921	0.806	1.327
Lingbot-VLA2	0.000316	0.310	0.539	0.002633	0.725	1.194

offline evaluation can serve as an efficient proxy for comparing manipulation models before costly real-robot deployment.

5.5 Real-Robot Experiments

We deploy the best-performing models from the four preceding tasks in a closed-loop system and evaluate their end-to-end performance on HEIR. Importantly, Lingbot-VLA2 achieves a subtask-level success rate of *71.79%* across *39* manipulation subtasks, while FastWAM achieves a success rate of *64.86%*. We therefore adopt Lingbot-VLA2 for episode-level evaluation. For navigation, InternVLA-N1 achieves a subtask-level success rate of *81.25%* across *32* evaluated navigation subtasks. Even so, when evaluated end-to-end at the episode level, combining the best-performing models across all components still yields a relatively low success rate of *18.52%* over the *27* evaluated episodes. This gap largely arises from error accumulation across key components of the HRI system, spanning perception, planning, and execution over long-horizon interactions, highlighting a persistent challenge for the community.

6 Conclusion

In this paper, we introduce HEIR, a dataset and benchmark for studying user-centric human–robot interaction in long-horizon, naturalistic household environments. HEIR jointly captures human egocentric video, gaze, and instruction together with robot observations and execution trajectories, with structured annotations linking human intent to planning and execution. We formulate a hierarchical evaluation spanning human intent understanding, subtask planning, navigation, and manipulation, and benchmark representative models through both offline and real-robot experiments. Our results demonstrate the value of harnessing human-centric intentional cues for natural HRI, while also revealing substantial error accumulation when these components are integrated for long-horizon execution. We believe HEIR provides an important empirical foundation for developing embodied AI systems that place humans at the center of robot understanding, planning, and interaction.

AI USE STATEMENT

Generative AI systems were used as research tools during the development of HEIR. As described in the paper and appendix, GPT-family models were used for VLM-assisted annotation refinement, where human-authored intent annotations, robot-view videos, and aligned interaction records were provided as inputs to generate structured subgoal descriptions and completion conditions. These model-generated outputs were used to enrich the benchmark construction process rather than replace human annotation. All AI-assisted annotations were manually reviewed and verified by the research team before being incorporated into the final benchmark.

Multimodal foundation models, including Gemma-family, Qwen-family, and InternVL-family models, were evaluated as benchmark participants for human intent understanding and planning tasks. These models were used only as experimental baselines and did not contribute to the creation of ground-truth annotations or evaluation labels.

Generative AI tools were also used to assist with manuscript proofreading and language refinement. The authors reviewed all AI-assisted outputs and take full responsibility for the dataset design, experimental methodology, analyses, claims, citations, and final manuscript.

ETHICS STATEMENT

This study involved human participants and real-world human–robot interaction recordings. All participants voluntarily participated in the study and provided informed consent before participation. Participants were informed about the data collection procedures, the intended research purpose, data usage, compensation, and their right to withdraw from the study.

HEIR collects multimodal human–robot interaction data, including egocentric video, gaze measurements, speech recordings, robot-view observations, and robot execution trajectories. Since first-person recordings in natural environments may contain privacy-sensitive information, including faces, voices, personal belongings, and surrounding scenes, privacy protection was carefully considered during dataset preparation. Publicly released benchmark samples will undergo de-identification procedures, including removal of direct identifiers and manual inspection of potentially sensitive content.

The dataset is intended solely for research on user-centric human–robot interaction and embodied AI. It is not designed for surveillance, unauthorized profiling, clinical diagnosis, or consequential decision-making about individuals. We encourage responsible use of HEIR, including obtaining user consent, minimizing unnecessary data exposure, protecting personal information, and evaluating potential limitations and failure modes of systems developed using this resource.

REPRODUCIBILITY STATEMENT

The paper and appendix provide detailed descriptions of the HEIR dataset construction, sensing hardware, synchronization procedures, annotation protocols, task formulations, evaluation metrics, model configurations, and experimental settings.

To facilitate reproducibility, we will release benchmark annotations, processed evaluation instances, task definitions, prompts, preprocessing scripts, evaluation code, and executable task-specific pipelines. For real-robot experiments, we additionally provide deployment configurations and evaluation protocols to facilitate reproduction of the reported results.

Due to the privacy-sensitive nature of continuous egocentric recordings and synchronized multimodal interaction streams, raw participant recordings cannot be fully released without restriction. Sensitive data will be made available through an appropriate controlled-access procedure and data-use agreement following privacy review. For experiments involving proprietary foundation models, we report model identifiers, prompts, and relevant inference settings, while noting that future changes to closed-source models may affect the exact reproducibility of model outputs.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

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

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 3674–3683. IEEE, 2018.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. In *International Conference on Learning Representations*, volume 2025, pp. 65205–65268, 2025.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847, 2021.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of machine learning research*, 24(240):1–113, 2023.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 653–660. IEEE, 2024.

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

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18995–19012, 2022.

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, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C.V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. 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)*, pp. 19383–19400, June 2024.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264. IEEE, 2018.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. *An introduction to statistical learning: with applications in R*, volume 2. Springer, 2021.

Chenxi Jiang, Chuhao Zhou, and Jianfei Yang. Rei-bench: Can embodied agents understand vague human instructions in task planning? In *International Conference on Learning Representations*, volume 2026, pp. 31706–31738, 2026.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. In *2026 International Conference on 3D Vision (3DV)*, pp. 499–509. IEEE, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, 2020.

Yuzhi Lai, Shenghai Yuan, Peizheng Li, Boya Zhang, Benjamin Kiefer, Tianchen Deng, and Andreas Zell. Fam-hri: Foundation-model assisted multimodal human-robot interaction combining gaze and speech. *IEEE Transactions on Automation Science and Engineering*, 2026.

Dongjun Lee, Juheon Choi, Dong Kyu Shin, Sinjae Kang, and Kimin Lee. Hierarchical policies from verbal and egocentric human signals for natural human-robot interaction. *arXiv preprint arXiv:2606.10276*, 2026.

Baiqi Li, Ce Zhang, Yu Fang, Yue Yang, Shangzhe Li, Mingyu Ding, and Gedas Bertasius. Watchact: A benchmark for behavior-grounded robot manipulation. *arXiv preprint arXiv:2606.26443*, 2026a.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pp. 80–93. PMLR, 2023a.

Chengyang Li, Kaiyi Xiong, Yuan Xu, Lei Qian, Yizhou Wang, and Wentao Zhu. Gazevla: Learning human intention for robotic manipulation. *arXiv preprint arXiv:2604.22615*, 2026b.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023b.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026c.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li E Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2024.

Kevin M Lynch and Frank C Park. Modern robotics. *Mechanics, Planning, and Control*, 2017.

So Yeon Min, Xavi Puig, Devendra Singh Chaplot, Tsung-Yen Yang, Akshara Rai, Priyam Parashar, Ruslan Salakhutdinov, Yonatan Bisk, and Roozbeh Mottaghi. Situated instruction following. In *European Conference on Computer Vision*, pp. 202–228. Springer, 2024.

Soroush Nasiriany, Sep Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations*, volume 2026, pp. 98643–98667, 2026.

Benjamin A Newman, Reuben M Aronson, Siddhartha S Srinivasa, Kris Kitani, and Henny Admoni. Harmonic: A multimodal dataset of assistive human–robot collaboration. *The International Journal of Robotics Research*, 41(1):3–11, 2022.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Juri Opitz. A closer look at classification evaluation metrics and a critical reflection of common evaluation practice. *transactions of the association for computational linguistics*, 12:820–836, 2024.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2017–2025, 2022.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

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

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL `https://qwen.ai/blog?id=qwen3.8`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8689–8696, 2020.

Rosario Scalise, Shen Li, Henny Admoni, Stephanie Rosenthal, and Siddhartha S Srinivasa. Natural language instructions for human–robot collaborative manipulation. *The International Journal of Robotics Research*, 37(6):558–565, 2018.

Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J Joshi, et al. Robovqa: Multimodal long-horizon reasoning for robotics. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 645–652. IEEE, 2024.

Snehesh Shrestha, Yantian Zha, Saketh Banagiri, Ge Gao, Yiannis Aloimonos, and Cornelia Fermüller. Natsgld: A dataset with speech, gesture, logic, and demonstration for robot learning in natural human-robot interaction. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 1093–1098. IEEE, 2025.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10737–10746. IEEE, 2020.

Jaehwi Song, Suchae Jeong, Byeongguk Jeon, Sungdong Kim, Minjoon Seo, Hyungmok Son, and Kimin Lee. Habit: Human-aware behavior and interaction training dataset for robot manipulation. *arXiv preprint arXiv:2606.31682*, 2026.

Tracey Yee Hsin Tay, Xu Yan, Jonathan Ouyang, Daniel Wu, William Jiang, Jonathan Kao, and Yuchen Cui. Intent at a glance: Gaze-guided robotic manipulation via foundation models. *arXiv preprint arXiv:2601.05336*, 2026.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

IN Team. Internvla-n1: An open dual-system vision-language navigation foundation model with learned latent plans. Technical report, Technical report, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

Yanming Wan, Jiayuan Mao, and Josh Tenenbaum. Handmethat: Human-robot communication in physical and social environments. *Advances in Neural Information Processing Systems*, 35: 12014–12026, 2022.

Yanming Wan, Yue Wu, Yiping Wang, Jiayuan Mao, and Natasha Jaques. Infer human's intentions before following natural language instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25309–25317, 2025.

Siyin Wang, Jinlan Fu, Feihong Liu, Xinzhe He, Huangxuan Wu, Junhao Shi, Kexin Huang, Zhaoye Fei, Jingjing Gong, Zuxuan Wu, et al. Roboomni: Proactive robot manipulation in omni-modal context. In *International Conference on Learning Representations*, volume 2026, pp. 30282–30314, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20213–20224. IEEE, 2023.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. In *International Conference on Learning Representations (ICLR)*, 2026.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, et al. From foundation to application: Improving vla models in practice. *arXiv preprint arXiv:2607.06403*, 2026.

Mohammad Samin Yasar, Md Mofijul Islam, and Tariq Iqbal. Posetron: Enabling close-proximity human-robot collaboration through multi-human motion prediction. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 830–839, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

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

Kuangji Zuo, Gen Li, Bofan Lyu, Yanshuo Lu, Boyu Ma, Shijia Han, Xinyu Zhou, Xichen Yuan, Chuhao Zhou, Jiaqi Bai, et al. Gaze2act: Gaze-conditioned vision-language-action policies for interactive robot manipulation. *arXiv preprint arXiv:2605.30282*, 2026.

A Additional Related Work

Human Intent Grounding for HRI. Grounding human intent through behavioral cues such as gaze and hand gestures has been extensively studied in both egocentric and exocentric vision. (Grauman et al., 2024) These complementary perspectives provide critical intentional cues that guide robots in grounding task-relevant intent across language, human behavior, and the surrounding scene. FAM-HRI aligns gaze with speech to ground referents across human and robot views, while GAMMA interprets gaze in scene context to infer manipulation goals and parameterize robot skills (Lai et al., 2026; Tay et al., 2026). RoboOmni extends the interaction context by integrating dialogue, speech, environmental sounds, and vision for implicit-intent reasoning (Wang et al., 2026). EDITH Lee et al. (2026) translates egocentric video, gaze, and speech into a queue of subtask instructions paired with grounding keyframes. Gaze2Act Zuo et al. (2026) maps human gaze to object masks and part-level locations in the robot view to condition a

VLA, with voice-triggered target selection enabling updates during execution. GazeVLA instead learns from human gaze supervision and predicts an intention representation before generating actions (Li et al., 2026b). Importantly, prior works focus on intentional cues for short-horizon tabletop interactions, whereas our work studies how to leverage human intent for long-horizon robot planning and execution in naturalistic environments.

Long-Horizon Robot Planning and Execution. Long-horizon assistance requires coordinating actions over changing physical states. ALFRED, BEHAVIOR-1K, and RoboCasa365 benchmark household tasks that compose navigation and manipulation (Shridhar et al., 2020; Li et al., 2023a; Nasiriany et al., 2026). RoboVQA adds temporally segmented human and robot videos with supervision for planning, progress assessment, and other video–text reasoning tasks (Sermanet et al., 2024). Interactive settings also make communication and human behavior part of task progression: TEACh interleaves dialogue with household execution, Situated Instruction Following (SIF) studies requests involving moving people and relocated objects, and PARTNR evaluates collaboration under spatial, temporal, and agent-capability constraints (Padmakumar et al., 2022; Min et al., 2024; Chang et al., 2025). Notably, long-horizon robot execution remains largely studied in simulation, while naturalistic HRI scenarios that require interleaved navigation and manipulation remain comparatively underexplored.

B HEIR Dataset Details

B.1 Data Collection and Hardware Setup

Collection protocol. Each session follows a household scenario involving successive requests for retrieval, handover, relocation, or cooperative manipulation. Across repetitions, the intended task goals remain consistent, while object positions within the same general areas, participants' wording, and interaction timing can vary. Participants wear Pupil Labs Neon glasses and look toward task-relevant objects and locations when issuing requests. A separate operator uses handheld controllers to teleoperate a Galaxea R1 Lite (2025 version) from a global scene view. The recordings retain the participant's subsequent verbal and nonverbal interaction, including reaching and individual handover requests.

Recorded streams. Table 7 summarizes the human- and robot-side recordings. Robot video provides complementary head and wrist views, while Parquet records contain arm, gripper, torso, and base states together with recorded actions. Measured gaze is associated with human scene-camera frames by timestamp and visualized on the egocentric video. Speech recordings preserve the participant's original utterances; their transcriptions and English translations are stored in the annotations.

Table 7: Human and robot data streams used for annotation. Video specifications refer to the recorded streams; gaze is matched to scene frames by timestamp.

Stream	Source	Recorded information
Human scene video	Pupil Labs Neon	1600×1200 RGB video; nominally 30 FPS
Human speech	Neon microphone	Audio synchronized with human video
Human gaze	Neon eye tracking	Timestamped measured gaze, overlaid on scene video
Robot head video	Two head cameras	1280×720 RGB video per camera; 30 FPS
Robot wrist video	Two wrist cameras	640×360 RGB video per camera; 30 FPS
Robot state–action	R1 Lite logs	Arm, gripper, torso, and base states; recorded actions in Parquet

Temporal synchronization. The human and robot cameras observe the same stopwatch at the beginning and end of a recording. Matching timer readings provide two corresponding frame pairs, (h_0, r_0) and (h_1, r_1), from which we obtain the linear mapping $r = r_0 + (h - h_0)(r_1 - r_0)/(h_1 - h_0)$. Here, h and r denote human and robot frame coordinates. The robot's 30 FPS timeline is the reference: robot videos and Parquet records are cropped together, and human video is resampled to the corresponding frame count. Human audio undergoes the same temporal scaling while preserving pitch. The observed pre-alignment drift is approximately 4 s over 10 min, corresponding to a 0.67%

duration adjustment. Final trimming removes the stopwatch observations and retains a common interaction interval with local frame indices shared by the aligned videos and robot records.

Navigation references. Timestamped wheel odometry and MapAnything visual estimates provide the planar base trajectories associated with navigation subtasks. Reconstructed scene geometry is registered to MID-360 scans, and trajectories are compared with HTC VIVE Tracker measurements from reference recordings. Section 3.4 reports the validation conditions and errors. These geometric references remain aligned with robot video and the execution annotations.

B.2 Hierarchical Annotation Protocol

Request evidence. Annotators work with aligned human and robot videos to identify request-level episodes and their speech intervals. Each request retains its original transcription and an English translation that preserves referential expressions. On the aligned 30 FPS timeline, the gaze evidence window begins 30 frames before speech onset, clipped at the recording start, and ends with the speech interval. Event boundaries use half-open frame intervals $[s, e)$; timestamps and corresponding Parquet indices are derived from this shared frame reference.

Grounded intent. Human annotators resolve the request into a task description, manipulated objects, goal entities, and a nominal navigation–manipulation plan. Object annotations record names, counts, colors, texture categories, and source locations. Goal entities identify intended recipients or destination references. Robot visibility is judged within the request's gaze evidence window. The nominal plan captures the sequence inferable before execution, including torso adjustments when required by the initial robot state. For references to earlier interactions, annotators use the preceding session context together with the current request evidence.

Execution structure. Annotators separately mark the robot's execution interval, outcome, and continuous navigation or manipulation subtasks. Navigation covers base travel and local repositioning; manipulation covers arm, gripper, and torso actions. A manipulation subtask can contain sequential actions involving multiple objects or both arms. The realized subtask sequence records the demonstrated execution and may differ from the nominal plan because of reachability or robot-configuration adjustments. Each subtask retains its identifier, type, and frame interval linking it to the corresponding video and state–action records.

VLM-assisted semantic enrichment. GPT-5.6 receives the human annotations, robot-view video, and aligned state–action records to generate a natural-language subgoal and completion condition for each annotated subtask. Subgoals describe the demonstrated objective and actions; completion conditions specify observable terminal states. Manipulation descriptions identify the objects, executing arms or grippers, and relevant torso adjustments. Navigation descriptions identify the destination and any objects that must remain held during travel. Human-defined subtask boundaries and execution order are preserved during enrichment. Intent refinement corrects grammar and spelling while retaining the annotated meaning. Human annotations and model-generated semantic fields have distinct roles: the former establish intent and execution structure, and the latter provide language supervision for subtask execution and progress assessment.

B.3 Quality Assurance and Review Procedures

Structural checks. The annotation interface checks required fields, unique episode identifiers, valid event intervals, temporal containment, and the navigation/manipulation type vocabulary. Speech, gaze, and robot execution intervals must lie within their episode, and each subtask interval must lie within the overall execution interval. The gaze window is checked against the speech boundaries. Reviewers use the recordings to assess semantic properties, including referent identity, robot visibility, demonstrated actions, and observable completion conditions.

Iterative semantic review. At each human-review round, we draw a new random sample of 20% of request-level episodes and inspect their intent annotations and all constituent subtask subgoals and completion conditions. The initial review identified eight ambiguous subtasks. These included minor base movements within segments annotated as manipulation, which required clearer descriptions of the observed motion. We refined the annotation rules and conducted VLM-assisted re-review, resulting in 34 annotation revisions. A subsequent random human review identified one remaining ambiguous subtask. After correcting it and further refining the rules, a targeted review identified

Table 8: Subtask annotations for episode E004. Types and intervals are human annotated; subgoals and completion conditions are VLM-enriched descriptions, condensed here from the annotated record. Intervals are in seconds on the synchronized session timeline.

Subtask / interval (s)	Subgoal	Completion condition
S001: Manipulation $[184.20, 191.57)$	Straighten the lowered torso while keeping both grippers empty and the base stationary.	Torso straightened; both grippers empty; base remains at the counter.
S002: Navigation $[191.60, 205.47)$	Return from the counter to the seated human with both grippers empty.	Base stopped facing the seated human; both grippers empty.
S003: Manipulation $[205.50, 238.50)$	Lower the torso; receive the stapler with the left gripper, then the tape roll with the right; retract both arms and straighten the torso.	Stapler held on the left and tape on the right; both arms retracted; torso straightened.
S004: Navigation $[238.50, 262.43)$	Navigate to the kitchen counter and white box while carrying the stapler in the left gripper and the tape roll in the right.	Base stopped at the counter; white box visible; both objects retained in their respective grippers.
S005: Manipulation $[262.47, 283.70)$	Lower the torso; place the stapler into the white box and retract the left arm, then place the tape roll into the same box and retract the right arm.	Both objects inside the box; both grippers empty and arms retracted; torso remains lowered.

two additional instances of the same issue, which were also corrected. The final random human review found no further issues. An additional review using Codex with the GPT-5.6 `ultra` setting likewise identified no further issues. This process combines repeated human inspection with rule-guided model review; the reported counts distinguish ambiguities found during inspection from revisions made during re-review.

B.4 Example of Human Annotation and VLM-Assisted Enrichment

Session context. An office-supply session contains five request-level episodes and 15 execution subtasks: six navigation and nine manipulation segments. The requests involve retrieving a stapler, holding tape while the participant cuts it, putting documents into a tray, returning the stapler and tape, and returning the robot to its starting location. We use the fourth episode, E004, to illustrate how human intent and execution annotations are enriched with subtask semantics.

Human annotation. The translated request is "Now please put the stapler and the roll of tape into the white box." Speech spans frames $[5334, 5477)$, and gaze evidence spans $[5304, 5477)$. The manipulated objects are a white stapler and a clear tape roll, both annotated as being in the participant's hands and not visible to the robot within this evidence window. The white box is the goal entity and is annotated as robot-visible. The grounded intent specifies straightening the torso, approaching the participant, receiving both objects, carrying them to the box, and returning them to it. The nominal plan and the recorded execution both follow manipulation–navigation–manipulation–navigation–manipulation.

Enriched subtask semantics. Table 8 pairs the five human-annotated execution intervals with their subgoals and completion conditions. The descriptions retain which arm holds each object, the carrying constraints during navigation, and the torso and gripper states at completion. For example, receiving both objects remains one manipulation subtask, whereas the subsequent navigation subtask explicitly retains both grasped objects. The final manipulation condition establishes that both objects are inside the box and the grippers are empty, linking the recorded actions to the requested outcome.

B.5 3D Point Cloud and Trajectory Comparison

The HTC VIVE Tracker is rigidly mounted on top of the robot. The blue trajectory denotes the fused output of MapAnything and wheel-base odometry, transformed into the robot base coordinate

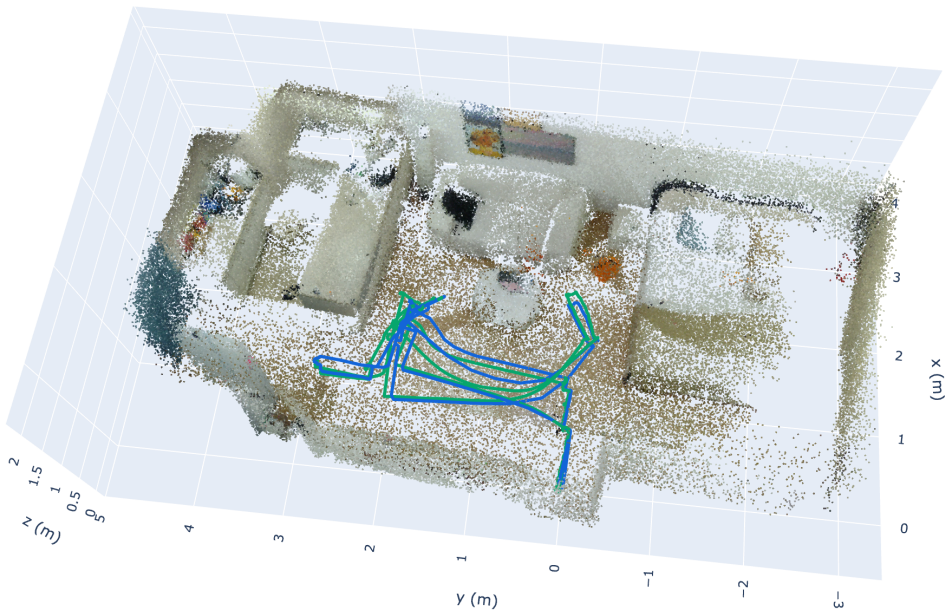

Figure 4: **Trajectory comparison between MapAnything–wheel-base fusion and HTC VIVE Tracker measurements.**

frame, while the green trajectory corresponds to the raw Tracker measurements without coordinate-frame alignment. Despite being expressed in different coordinate frames, the two trajectories exhibit strong consistency in their motion patterns: they remain approximately parallel during straight-line motion and show closely synchronized heading changes during turns.

C System Overview

C.1 Hierarchical Execution Pipeline

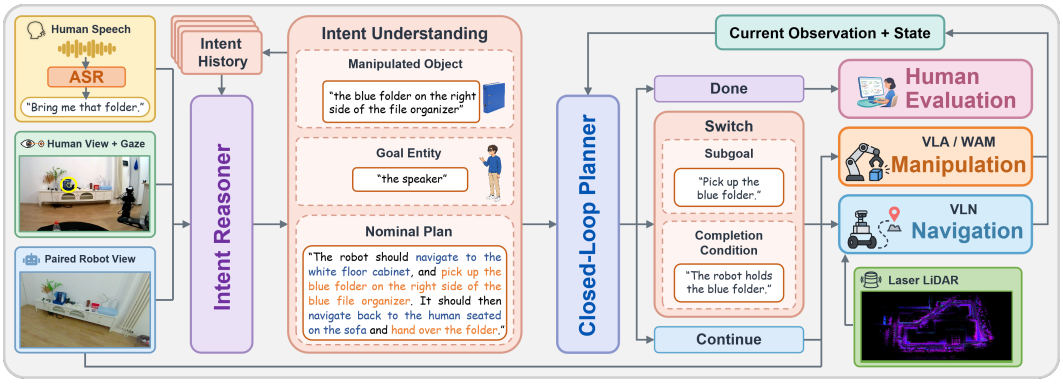

Figure 5: **Real-robot execution pipeline of HEIR.** For the request "Bring me that folder.", the intent reasoner infers a structured intent, and the closed-loop planner then dispatches subtasks to the navigation and manipulation backends until DONE.

Algorithm 1 HEIR real-robot evaluation pipeline. Intent understanding initializes each episode, while a closed-loop planner coordinates navigation and manipulation.

Input: Live streams $X_k^H = (V_k^H, A_k^H, G_k^H)$ and X_k^R, LiDAR stream L_k, predicted intent history $\mathcal{H}_{<k}$, intent reasoner $\mathcal{R}$, planner $\mathcal{P}$, navigation policy π_{nav}, manipulation policy π_{manip}, inactivity threshold τ_{idle}
Initialize: Subtask type $m \leftarrow \varnothing$, subgoal $g \leftarrow \varnothing$, completion condition $c \leftarrow \varnothing$
$u_k \leftarrow$ WAITFORASRREQUEST(A_k^H) ▷ Start episode k
$(\widetilde{V}_k^H, \widetilde{G}_k^H, \widetilde{V}_k^R) \leftarrow$ GETREQUESTCONTEXT(X_k^H, X_k^R)
$\hat{z}_k \leftarrow \mathcal{R}(u_k, \widetilde{V}_k^H, \widetilde{G}_k^H, \widetilde{V}_k^R, \mathcal{H}_{<k})$
$\mathcal{H}_{\le k} \leftarrow \mathcal{H}_{<k} \,\|\, [\hat{z}_k]$ ▷ Append the predicted intent
while true do
  $(o_{k,t}^R, s_{k,t}) \leftarrow$ READLATESTROBOTCONTEXT(X_k^R)
  $(d, m', g', c') \leftarrow \mathcal{P}(\hat{z}_k, o_{k,t}^R, s_{k,t}, g, c)$
  if $d =$ DONE **then**
    STOPROBOT
    break ▷ Terminate episode k
  else if $m = \varnothing$ **or** $d =$ SWITCH **then**
    $(m, g, c) \leftarrow (m', g', c')$ ▷ Initialize or switch subtask
  else
    Retain (m, g, c) ▷ CONTINUE
  end if
  if $m =$ NAVIGATION **then**
    RUNORCONTINUEBACKEND($\pi_{\text{nav}}, g, X_k^R, L_k$)
  else
    RUNORCONTINUEBACKEND($\pi_{\text{manip}}, g, X_k^R$)
  end if
  WAITFORINACTIVITY($X_k^R, \tau_{\text{idle}}$) ▷ Monitor robot states and actions
end while
return $\mathcal{H}_{\le k}$

Fig. 5 and Algorithm 1 show how the four tasks of Section 4 are composed into a single system on the physical robot. The system listens continuously, and each spoken request opens a new episode k. The intent reasoner $\mathcal{R}$ (Appendix C.2) is invoked once per request to infer a structured intent $\hat{z}_k$, which is appended to the intent history. Execution then proceeds in an event-triggered closed loop: the planner $\mathcal{P}$ (Appendix C.3) reads the latest robot observation and state and decides whether to CONTINUE the active subtask, SWITCH to a new subtask specified by a type m, a subgoal g, and a completion condition c, or end the episode with DONE. The subtask type selects the navigation or manipulation backend (Appendix C.4), which executes g until the robot becomes idle, at which point the planner is queried again. After the episode ends, the updated history is carried over to the next request, and human evaluators assess both the executed subtasks and the outcome of the whole episode; DONE thus marks the end of execution rather than success.

C.2 INTENT REASONER

Model. The real-robot system uses Qwen3.8-27B (Qwen Team, 2026b) as the intent reasoner. Although Gemma-4-31B performs slightly better offline (Section 5), Qwen3.8-27B achieves comparable grounding performance (Appendix E) and better meets the real-time and latency requirements of live interaction, for which we had already optimized its inference (Appendix D). It is not fine-tuned on HEIR; instead, it is prompted to produce the structured intent directly (Appendix D.2).

Inputs and outputs. For episode k, the ASR transcript u_k triggers the reasoner, and the request-aligned context is retrieved from the live streams: human egocentric frames $\widetilde{V}_k^H$ overlaid with the measured gaze $\widetilde{G}_k^H$, together with the corresponding robot views $\widetilde{V}_k^R$. The reasoner also receives $\mathcal{H}_{<k}$, the intents it predicted for earlier episodes in the same session, which corresponds to the predicted-history setting of the offline evaluation. It outputs a structured intent $\hat{z}_k$ that follows the HEIR intent annotation schema (Section 3.1): a grounded task description, the manipulated objects, the goal entities, and a nominal navigation–manipulation plan. In the example of Fig. 5, gaze grounds the deictic request "Bring me that folder." to the blue folder on the right side of the file organizer, and the goal entity resolves to the speaker.

Role in execution. The reasoner runs once per request, before any robot motion. Its nominal plan is not executed step by step; it only outlines the expected subtask structure and serves, together with the grounded objects and goal entities, as task context for the planner, which decides the realized subtasks online from the robot's state.

C.3 Closed-loop Planner

Model. The planner is Qwen3.5-4B (Qwen Team, 2026a), fine-tuned on decision-level examples constructed from HEIR's annotated execution segments with their subgoals and completion conditions; each example is restricted to the observations and history available at its decision point (Appendix D.1). We select it for two reasons. First, with the predicted intent as additional input, it achieves the highest planning accuracy and macro-F1 among the compared planners (Tab. 4). Second, its small size keeps the latency of each decision low in a loop that is queried repeatedly within an episode.

Inputs and outputs. At each planning step t, the planner receives the structured intent $\hat{z}_k$, the latest robot observation $o^R_{k,t}$ and state $s_{k,t}$, and the active subgoal g with its completion condition c (empty at the first step). It returns a decision $d \in \{\textsc{Continue}, \textsc{Switch}, \textsc{Done}\}$ and a candidate subtask (m', g', c'). At the first step or upon SWITCH, the candidate becomes the active subtask, *e.g.*, $m =$ MANIPULATION, $g =$ "Pick up the blue folder.", and $c =$ "The robot holds the blue folder."; under CONTINUE, the active subtask is retained; and DONE stops the robot and ends the episode.

Event-triggered invocation. The planner is not queried at a fixed rate, so that a running backend is not interrupted in the middle of a motion. Instead, the system monitors the robot's states and actions and re-queries the planner once the robot has been inactive for $\tau_{\text{idle}} = 20\,\text{s}$. The robot is considered inactive if, throughout this window, the mobile base moves by less than $0.01\,\text{m}$ and all torso and arm joints remain within $0.05\,\text{rad}$ of their positions at the start of the window. Coming to rest only triggers a new decision: the backend may have completed the subgoal, stalled, or failed, and the planner judges completion against c from the current observation and state.

C.4 Navigation and Manipulation Backends

Selection criteria. Both backends receive the active subgoal g as a natural-language instruction, so any navigation or manipulation policy that accepts such instructions can be plugged into the pipeline. For the real-robot system, we deploy, for each subtask type, the model that performs best in the corresponding stand-alone offline evaluation (Section 5). For manipulation, candidates must additionally support bimanual control and torso lifting, which many HEIR subtasks require to reach objects at different heights or to use both arms. Both deployed backends are fine-tuned on the HEIR training split.

Navigation. Navigation subtasks are executed by InternVLA-N1 (Team, 2025; Wei et al., 2026), fine-tuned on HEIR navigation subtasks paired with their subgoal descriptions. It achieves the best action prediction and stopping performance in the offline evaluation (Tab. 5). At run time, it receives g and the robot-view stream X^R_k and predicts navigation actions for the mobile base, while the LiDAR stream L_k from the Mid-360 sensor is used for collision avoidance.

Manipulation. Manipulation subtasks are executed by Lingbot-VLA2 (Wu et al., 2026), fine-tuned on HEIR manipulation subtasks with their subgoal descriptions as instructions. It achieves the lowest joint-space and end-effector errors for both first-step and action-chunk prediction among the compared VLA and WAM models (Tab. 6). At run time, it receives g and X^R_k and outputs action chunks that control both arms, both grippers, and torso lifting.

D Training and Implementation Details

D.1 Foundation Models and Training Configuration

Intent reasoner. We use the pretrained Gemma-4-31B, Qwen3.8-27B, and InternVL3.5-38B models without fine-tuning. Their behavior is specified through the intent-understanding prompt and supplied context. Input-modality and history comparisons change the inference inputs while retaining the model weights.

Planner supervision. We directly construct supervised planner examples from HEIR's annotated execution sequences. Each example pairs observations and task context with one of three decisions: `CONTINUE` maintains the current subtask, `SWITCH` advances to the next subtask, and `DONE` ends the episode when its goal is complete. SWITCH examples additionally supervise the next subtask's type, subgoal, and completion condition. These dataset-derived examples support learning both progression decisions and the objectives passed to execution backends. Observations and input history are restricted to information available at the decision time; execution annotations supply the supervision targets.

The planner training pool contains 492 unique observations from 99 episodes and 97 recording groups: 238 CONTINUE, 223 SWITCH, and 31 DONE observations. Each epoch retains these observations and repeats samples to obtain 400 examples per class, giving 1,200 exposures. For CONTINUE and SWITCH, each class contains 200 navigation and 200 manipulation exposures; all 400 DONE exposures are manipulation examples. Additional draws assign weight 2 to observations in temporal pairs and weight 1 to other observations, subject to these quotas. The development set contains 192 queries and receives no gradient updates.

Planner optimization. Cosmos3-Nano, Qwen3.5-4B, and Gemma-4-E4B-it each start from their pretrained base weights with newly initialized LoRA adapters. For each backbone, instruction-only and instruction-plus-intent runs share initialization settings, sampling, and the training objective. LoRA uses rank 16, alpha 32, and dropout 0.05; base-model and visual parameters are frozen. We use BF16 base weights, FP32 adapters, and AdamW with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and weight decay 0.01. The global gradient-norm limit is 1.0. The effective batch is 32 observations, except for the last update of each epoch, which contains 16. Each epoch therefore has 38 optimizer updates. Linear warmup occupies 5% of the planned updates and is followed by cosine decay. Table 9 gives model-specific settings.

Table 9: Planner training configuration. Settings are shared by the two input variants of each backbone. Updates count optimizer steps.

Setting	Cosmos3-Nano	Qwen3.5-4B	Gemma-4-E4B-it
Peak learning rate	2×10^{-6}	10^{-6}	10^{-6}
GPUs per input variant	4	8	4
Configured accumulation	8	4†	8
Completed epochs / updates	2 / 76	2 / 76	1 / 38
Selected epoch / updates	2 / 76	1 / 38	1 / 38
Warmup updates	4	4	2
LoRA target modules	144	152	132
Trainable parameters	15,335,424	14,376,960	9,076,736

†Qwen schedules singleton and paired units dynamically. A pair requires two sequential forwards, so 4 is not a fixed forward-call count; the effective observation batch remains 32, or 16 at the epoch tail.

Only assistant-answer tokens receive supervision. Decision tokens and the remaining answer tokens are averaged separately; the decision-token weights are 2, 2, and 4 for Cosmos, Qwen, and Gemma, respectively. Qwen additionally uses three-class cross-entropy with weight 1 and a temporal ranking loss with weight 0.5. For a same-subtask pair containing an earlier CONTINUE observation and a SWITCH or DONE endpoint, the ranking term encourages the endpoint-minus-CONTINUE logit margin to increase by at least 1. Its softplus penalty is summed over pairs and normalized by the batch observation count. The selected Qwen epoch contains 106 pair units covering 59 phases: 84 CONTINUE–SWITCH and 22 CONTINUE–DONE pairs. Cosmos and Gemma use answer-token supervision without these auxiliary terms. All runs use seed 20260924, with process-local seeds offset by rank.

Cosmos uses the second-epoch checkpoint, fixed before training the paired conditions. Qwen and Gemma select a common checkpoint epoch for both input variants using their mean development balanced accuracy, with content and false-DONE safeguards; ties use mean accuracy and then content F1. Qwen selects epoch 1 from a run completed for 2 epochs, while Gemma completes and selects epoch 1. Qwen's selected checkpoint consequently contains 38 updates from a schedule planned for 76. Test predictions are excluded from checkpoint selection.

Navigation training. We adapt the released NaVid, StreamVLN, and InternVLA-N1 checkpoints using LoRA with rank 32, alpha 64, and dropout 0.05. Original model parameters and visual encoders are frozen, including InternVLA-N1's System 1 trajectory module. Adapters target attention and feed-forward projections in the language decoder. NaVid has 79,953,920 trainable parameters; StreamVLN and InternVLA-N1 each have 80,740,352. Adapter parameters are FP32, while decoder computation uses FP16 with gradient scaling for NaVid and BF16 for the other models. The session split contains 180 training, 38 validation, and 60 testing sessions.

Initial action adaptation uses 17,137 training and 3,443 validation examples with valid four-action labels. Movement labels are derived from approximately one second of future reconstructed base motion; future motion supplies targets only. NaVid and InternVLA-N1 also receive window-level target-visibility supervision, comprising 1,938 training and 407 validation examples. Each epoch uses two exposures per visibility example with coefficient 0.5 and within-task class balancing. This auxiliary task predicts Yes/No from robot-camera observations and a target name.

Table 10 summarizes the training stages and tested checkpoints. All stages use effective batch 96, AdamW with weight decay 0.01, approximately 3% linear warmup followed by cosine decay, and gradient clipping at 1.0. Optimizers and schedulers are reinitialized between stages. The final termination-aware stage uses pose- and wheel-feedback-constrained STOP targets. A terminal interval must remain within 2 cm and 3 degrees of the final phase pose, with all three wheel-speed magnitudes at most 0.01 m/s; STOP supervision begins after approximately 0.2 s of confirmed dwell. The thresholds are fixed before testing. Training and validation contain 36,072 and 7,572 termination-scored queries, including 2,584 and 532 STOP positives, respectively.

Table 10: VLN adaptation stages and selected checkpoints. Epoch counts refer to individual stages; each later stage starts from the selected checkpoint of the preceding applicable stage.

Setting	NaVid	StreamVLN	InternVLA-N1
Initial action adaptation	3 epochs; 10^{-4}	3 epochs; 10^{-4}	3 epochs; 10^{-4}
Selected action checkpoint	Epoch 2	Epoch 3	Epoch 3
Joint action/visibility adaptation	3 epochs; 2×10^{-5}	Not applied	3 epochs; 2×10^{-5}
Selected joint checkpoint	Epoch 3	—	Epoch 3
Termination-aware adaptation	1 epoch; 10^{-5}	1 epoch; 10^{-5}	1 epoch; 10^{-5}
Adapter used for testing	Joint epoch 3	Action epoch 3	Termination-aware epoch 1

Termination-aware training adds a binary STOP loss with coefficient 0.25 to answer-token cross-entropy. Queries without a valid four-action label receive only binary supervision; visibility examples receive no STOP loss. The STOP logit is compared with the log-sum-exp of unique movement-answer first-token logits. Added STOP examples within each phase share total sample weight 0.25. A new checkpoint is accepted only if validation STOP precision improves, recall drops by at most 2 percentage points, first-STOP interval hit rate does not decline, and action macro-F1 and applicable visibility macro-F1 each drop by at most 1 percentage point. NaVid and StreamVLN retain their earlier adapters; InternVLA-N1 passes the safeguards and uses the new adapter. These choices are frozen before testing.

Manipulation training. All six VLA/WAM models are fine-tuned from pretrained weights for 5 epochs. An epoch is one pass through the model-specific training-window dataset. The runs use 198 training sessions and 20 validation sessions, annotated subgoals as conditioning text, and a 30 Hz source action stream. Most runs use manipulation phases; Lingbot-VLA2 additionally uses navigation phases. Normalization statistics use training data only. Table 11 reports effective global batches, native chunk lengths, and optimization settings. The random seed is 42.

The π_0, $\pi_{0.5}$, and π_0-FAST models initialize from their released base checkpoints. They train LoRA parameters together with unfrozen vision/projection parameters under their model-defined freeze filters. The first two use LoRA in both the VLM and action expert; FAST uses VLM LoRA. Computation is BF16 without EMA. FastWAM initializes from its released three-camera checkpoint, trains video/action DiT and proprioception parameters, and freezes its VAE and text encoder; its three torso action channels are zero-initialized. It uses BF16, ZeRO stage 1, activation checkpoint-

Table 11: VLA/WAM fine-tuning configuration. All models train for 5 epochs from pretrained weights. Batch denotes the effective global batch; chunk denotes the native action sequence length.

Model	Batch	Chunk	Optimizer	Peak LR	Weight decay
π_0	960	50	AdamW	2.5×10^{-5}	10^{-10}
$\pi_{0.5}$	1344	50	AdamW	2.5×10^{-5}	10^{-10}
π_0-FAST	208	32	AdamW	2.5×10^{-5}	10^{-10}
FastWAM	2816	32	AdamW	2.5×10^{-5}	0.01
DreamZero	32	24	AdamW	10^{-5}	10^{-5}
Lingbot-VLA2	256	50	Muon/AdamW	10^{-5}	0

The π models, FastWAM, and DreamZero use 5% warmup and cosine decay; the π models decay to 2.5×10^{-6}. Lingbot uses 1% warmup followed by a constant rate. All use gradient-norm limit 1.0. AdamW betas are $(0.9, 0.95)$ for the π models and FastWAM, and $(0.9, 0.999)$ for DreamZero. Lingbot retains its native parameter-group-specific optimizer rules.

ing, and 352 samples per GPU on 8 GPUs. DreamZero initializes from DreamZero-AgiBot and trains rank/alpha $4/4$ LoRA plus the state/action encoders and action decoder. Frozen parameters are BF16; trainable parameters and optimizer states are FP32. Lingbot-VLA2 uses the 20D configuration and native full-model post-training, including its vision encoder, MoE, and auxiliary supervision. It uses microbatch 32 on 8 GPUs, FSDP2, and the native FP32 setting.

D.2 Inference Settings and Prompt Design

Intent prompt and evidence. The intent reasoner is queried once before each episode's execution. The prompt supplies the original instruction, chronologically ordered request-window images, and available pre-execution state. Paired images place the human view on the left and four robot views on the right, arranged as head-left/head-right above wrist-left/wrist-right. Measured gaze overlays provide temporal attention cues. The prompt instructs the model to resolve entities across views using appearance and spatial context and to assess robot visibility over the supplied request window.

The output is one JSON object with six fields: `human_intent`, `manipulated_object_count`, `manipulated_objects`, `goal_entity_count`, `goal_entities`, and `full_task`. Object entries include names, color/texture/source-location attributes, and robot visibility; goal entries identify destinations or recipients and their visibility. Counts equal array lengths. The nominal plan contains only navigation and manipulation labels, with torso adjustments included in manipulation. A fixed scene-entity vocabulary standardizes English names across requests. Six annotation-format examples illustrate the schema and phrasing without supplying example images. The prompt requires evidence-based grounding for the current request rather than copying example attributes or plans.

History is supplied separately as earlier instructions paired with either human annotations or the same model's prior raw intent predictions. No-history conditions supply an empty array. The prompt distinguishes earlier intended actions from evidence that those actions were completed, and prioritizes current observations over historical attributes. When available, measured torso-posture facts are interpreted with their calibration and transition status. The model returns the six fields directly, without explanatory text or an outer annotation wrapper.

Planner inputs and outputs. Each training example uses the English instruction, current phase, a 26-channel robot state, and one synchronized 1280×768 RGB panel. The upper row contains the two head views and the lower row the two wrist views. The intent condition additionally supplies annotated `human_intent`, enabling an oracle-intent component comparison. Cosmos receives completed-phase history containing type, subgoal, and completion condition; Qwen and Gemma receive completed-action history containing type and executed subgoal. Training uses GT history with probability 1 and no history dropout. These representations are matched between the two input variants of each backbone.

Cosmos and Qwen use image-processing budgets of 65,536–983,040 pixels; Gemma uses a maximum of 1,120 soft image tokens. Prompt plus answer must fit within 16,000 tokens, checked without

truncation. Training applies no online image augmentation, and thinking is disabled in the chat template. All models share the output contract: `decision` is CONTINUE, SWITCH, or DONE, and `next_subtask` is null for CONTINUE and DONE. For SWITCH, it contains `phase_type`, `subgoal`, and `completion_condition`.

Navigation interface. The VLN decoder receives the navigation subgoal with current and historical robot imagery and predicts FORWARD, LEFT, RIGHT, or STOP through its native answer format. Auxiliary visibility examples retain model-specific sampling: approximately 4 Hz for NaVid and at most 8 paired-view time points for InternVLA-N1. Wheel feedback, final phase poses, and future frames are used for label construction rather than navigation-model inputs. Offline input comparisons reuse the same selected weights for subgoals, original instructions, and annotated intents; the main before/after fine-tuning comparison uses subgoals.

Manipulation representations. VLA/WAM inputs include the current robot state and head, left-wrist, and right-wrist images, processed with each model's native image pipeline and prompt wrapper. The π models and FastWAM use 17 active state/action dimensions: 12 arm joints, 2 grippers, and 3 torso joints. DreamZero observes 17 dimensions but outputs 14 arm/gripper dimensions. Lingbot-VLA2 adds body-frame base velocities to obtain 20 active dimensions, using a 55-slot internal representation with unused action slots masked. The π_0 and $\pi_{0.5}$ interfaces pad active state/action vectors to 32; DreamZero pads state to 64 and action to 32.

FastWAM predicts absolute positions. The π models, DreamZero, and Lingbot-VLA2 predict arm commands relative to the current measured state, with absolute gripper commands. Where torso actions are modeled, the π and Lingbot targets are next-frame measured torso positions relative to the anchor state, while FastWAM uses absolute next-frame positions. Lingbot base targets are next-frame wheel-feedback-derived body velocities. These future measurements are supervision targets. The native normalization is mean/std for π_0, FastWAM, and Lingbot; quantile-based for $\pi_{0.5}$ and FAST; and clipped min/max scaling for DreamZero.

Native action chunks contain 50, 50, 32, 32, 24, and 50 commands for π_0, $\pi_{0.5}$, FAST, FastWAM, DreamZero, and Lingbot, respectively. Offline evaluation compares the first action and a common 24-command horizon. Text-token limits are 64, 200, and 256 for the π family, 128 for FastWAM, and 72 for Lingbot. FastWAM supervises 9 video frames at $t, t+4, \ldots, t+32$; DreamZero uses 9 at $t, t+3, \ldots, t+24$ with a 352×640 multi-view layout. Their padded tails are masked. The π training windows remain within manipulation-phase boundaries. Original-instruction and intent conditions change inference-time language inputs to the same trained models.

D.3 Real-Robot Deployment Details

We deploy the hierarchy on the Galaxea R1 Lite mobile dual-arm robot. InternVLA-N1 is the selected navigation backend, based on its performance in the evaluated setting, and Lingbot-VLA2 supplies manipulation actions. The deployed InternVLA-N1 uses the termination-aware checkpoint specified above. Its navigation adapter translates the model's semantic actions into robot-base commands, with MID-360 LiDAR providing collision avoidance. The manipulation adapter converts Lingbot outputs to the robot's action interface using the state/action conventions described above.

At the start of each episode, the intent reasoner interprets the request and supplies its predicted structured intent to the planner. The planner uses updated robot observations and state to continue the active subtask, switch backends with a new subgoal and completion condition, or terminate the episode. Completed execution and current measurements provide the runtime context; GT history and annotated intent are the separately specified conditions of component training and evaluation. Algorithm 1 specifies the dispatch and inactivity-monitoring logic. The 30 Hz recording rate and 24-command offline horizon describe the data and evaluation protocol; online execution follows each backend's control interface.

E Additional Experimental Results

Tab. 12 and 13 reinforce the same conclusion across Qwen3.8-27B and InternVL3.5-38B. Input ablations show that language alone is insufficient for reliable intent understanding: adding human or robot visual context substantially improves grounding and planning, while gaze provides additional

Table 12: *Input ablations for Qwen3.8-27B and InternVL3.5-38B.* All scores are scaled by 100; ↑ denotes higher is better. **Bold** indicates the best score across input conditions for each model. RV is N/A without robot views.

	Grounding		Planning	Visibility	Text
Input condition	Slot F1 ↑	Loc. Acc. ↑	Plan EM ↑	RV F1 ↑	BERTScore F1 ↑
Qwen3.8-27B					
Textual Instruction only	61.36	24.67	37.57	N/A	92.97
Robot View only	73.58	60.89	61.62	**63.64**	93.14
Human View only	75.56	67.98	68.65	N/A	93.17
Human + gaze	77.38	74.02	**74.59**	N/A	**93.18**
Human + Robot	76.25	71.13	71.08	58.43	93.08
Human + Robot + gaze (full)	**77.94**	**75.59**	73.24	62.45	93.11
InternVL3.5-38B					
Textual Instruction only	56.70	30.97	30.00	N/A	92.81
Robot View only	62.81	50.39	63.51	**48.78**	92.90
Human View only	65.79	57.48	66.49	N/A	**92.92**
Human + gaze	66.15	**61.68**	64.59	N/A	92.86
Human + Robot	67.57	57.48	**68.65**	44.97	92.79
Human + Robot + gaze (full)	**68.23**	59.84	65.95	43.90	92.71

gains, particularly for localization and plan prediction. The strongest configuration varies slightly by metric, but combining multimodal human-centric context generally yields the best overall grounding performance.

Table 13: *Intent understanding results for Qwen3.8-27B and InternVL3.5-38B under different history conditions.* All scores are scaled by 100; ↑ denotes higher is better. **Bold** indicates the best score within each model, excluding ground-truth history.

	Grounding		Planning	Visibility	Text
History	Slot F1 ↑	Loc. Acc. ↑	Plan EM ↑	RV F1 ↑	BERTScore F1 ↑
Qwen3.8-27B					
Ground-truth history	81.92	80.05	78.38	66.34	93.64
No history	77.72	74.28	73.51	**64.82**	93.17
Predicted history	**78.91**	**78.22**	**76.22**	64.69	**93.27**
InternVL3.5-38B					
Ground-truth history	71.83	67.72	69.73	47.51	93.05
No history	68.00	60.10	65.41	42.19	92.55
Predicted history	**68.69**	**62.47**	**65.68**	**45.09**	**92.59**

History context ablations further show that intent understanding benefits from cross-episode context. Predicted history improves most grounding, planning, visibility, and text metrics over the no-history setting, while ground-truth history provides a clear upper bound. The consistent trend across both model families suggests that successive HRI episodes are interdependent and that modeling interaction history is important for long-horizon intent understanding.