INTERACTIVEWORLD: A VERIFIABLE HUMAN-CENTRIC ENVIRONMENT FOR BUILDING BETTER INTERACTIVE MLLM ASSISTANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

People have long dreamed of an assistant that is always there to guide them and remind what they forget, from new dishes to lost keys. Powered by multimodal large language models (MLLMs), egocentric AI assistants on wearable devices are bringing this dream within reach: they see what the user sees and give instructions for user's physical work when necessary. However, testing such assistance is challenging and usually requires real people. The expensive and rare real interaction data leaves a key question unanswered: ***Can today's MLLMs truly help people in physical world?*** To answer this question, we propose **InteractiveWorld**, the first verifiable environment for closed-loop interaction between human and AI assistant. Unlike existing robot-centric simulators, InteractiveWorld lets the assistant guide a human agent who observes the scene, understands instructions, and acts independently. It thus realistically simulates how AI assistant collaborates with human: the assistant instructs, the human acts, and the environment executes actions and updates what they see next. Building on it, we introduce **InteractiveBench**, the first dynamic benchmark for egocentric AI assistants, with 120 multi-turn tasks across six categories. In each task, the assistant guides the human in real time, and every instruction shapes the outcome, which is scored by verifiable, state-based rewards. Experiments with 12 frontier MLLMs show they are still far from reliable assistants. Even the strongest model, Claude Fable 5.1, achieves at most 20% and 25% on long-horizon Guidance and Planning tasks, respectively. Overall, our work offers a verifiable platform and a rigorous evaluation set, laying a solid foundation for research on interactive MLLM assistant that truly help people in physical world.

1 INTRODUCTION

A long-standing goal of AI is to help people in the physical world. In everyday tasks, people need a companion that sees what they see, remembers what they forget, and tells them what to do next. ***Egocentric AI assistants*** aim to fill this role: running on wearable devices, they share the user's first-person view and guide users in natural language. Unlike robots, they avoid physical control: the human acts, while the assistant focuses on perception, memory, reasoning, and communication. As MLLMs rapidly improve (Anthropic, 2026a; OpenAI, 2026b), it makes egocentric AI assistants a practical, near-term path to helping people in the physical world.

However, evaluating whether assistants truly help people remains challenging. Real-world experiments are costly and hard to scale with human. Moreover, existing benchmarks are almost all ***static***: a model watches a pre-recorded first-person video and answers questions about it (Grauman et al., 2022; Cheng et al., 2024a) or predicts when and how to intervene (Wang et al., 2023a; Zhang et al., 2025a). Whatever the model says, the video remains unchanged. On the contrary, real-world assistance is ***dynamic***: the assistant and human interact over many turns with diverse rewards, requiring abilities that fixed videos cannot test. **(1) Closed-loop adaptation.** The assistant must continually adapt its guidance to the evolving scene and human responses, since its instructions directly change the human's actions and what it sees next. **(2) Compositional capabilities.** The assistant must combine skills such as tracking object states, remembering events, and communicating clearly, whereas static benchmarks typically test one skill at a time. This leads to our core question: ***How can we faithfully evaluate an assistant's ability to interact with and help humans in real world?***

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: (Top) **Interaction Loop in InteractiveWorld.** The Assistant agent provides guidance based on current observations and history, which the Human agent grounds into executable actions to update the environment.(Bottom) **Example task from InteractiveBench.** The Assistant helps the human choose dinner (using a daily plan and food labels) and guides its placement on the table. Interaction relies on natural language and is evaluated by state-verifiable metrics.

To answer this question, we propose **InteractiveWorld**, to the best of our knowledge the first environment for closed-loop interaction between humans and assistants. Built on Isaac Sim (NVIDIA, 2025), it contains over 8,000 objects and 50 scenes. As shown in Fig. 1 (top), interactive loop in InteractiveWorld consists of three components: (1) an *Assistant agent* that, like a real wearable assistant, shares the human's egocentric view, receives the task from the human, and decides when and how to offer natural-language guidance; (2) a *Human agent*, driven by an MLLM, that interprets the guidance based on its observations and converts it into high-level commands from 36 action primitives (e.g., `[Grab]<cup>`), much as a person translates advice into behavior; and (3) an *Interactive Environment* that stably executes these commands through a full-body human model capable of autonomous navigation, posture adaptation, and collision-free manipulation, while continuously rendering the resulting first-person view. Through this closed loop, ***InteractiveWorld realistically simulates how an assistant helps a person with everyday tasks***.

Building on InteractiveWorld, we introduce **InteractiveBench**, the first closed-loop dynamic benchmark for egocentric AI assistants, comprising 120 human-reviewed tasks across six categories: *Guidance, Planning, Accessibility, Decision, Prevention, and Finding*. Each task requires the assistant to guide the human over many turns by combining perception, memory, reasoning, and communication. Task success is verified automatically from the final object states in the simulator. Since the assistant's guidance directly shapes the human's actions for the task, InteractiveBench can faithfully exam whether a model helps human. *We also provide several demos in the supplementary material.*

Experiments with 12 frontier MLLMs on InteractiveBench show ***state-of-the-art models still have substantial room for improvement in interactive assistance.*** Even the strongest model, Claude Fable 5.1, succeeds on only 41.7% of the tasks, and all other models stay at or below 33.3%. In particular, on Guidance and Planning, the best models reach only 20.0% and 25.0%, respectively, indicating models still perform poorly on long-horizon interactive tasks. We further analyze model behaviors in detail and distill five key findings in Sec. 4.1. In summary, our contributions are as follows:

- We present **InteractiveWorld**, the first verifiable human-centric interaction environment, in which an assistant's natural-language guidance drives a human agent's actions and changes the world, enabling realistic simulation of their everyday collaboration.
- We introduce **InteractiveBench**, the first multi-turn, closed-loop benchmark for evaluating models as egocentric AI assistants in the physical world, where assistant outputs shape task outcomes and success is measured by verifiable, state-based rewards.
- Experiments reveal major gaps in current MLLMs' dynamic interaction. Our analyses identify perception, memory, and communication as key bottlenecks. *These findings underpin future interactive MLLM research, with InteractiveWorld providing a verifiable platform.*

2 INTERACTIVEWORLD

In this section, we introduce InteractiveWorld, the first closed-loop human–assistant interaction environment providing a realistic, verifiable platform for evaluating egocentric AI assistants.

Challenges. Existing simulators provide robust physics and rich 3D assets for embodied AI. However, they are robot-centric, focusing on low-level motion planning and direct joint control. Transitioning to egocentric AI assistants introduces a paradigm shift: ***the agent must guide human via natural language rather than directly manipulating hardware.*** Simulating interaction between human and assistant requires modeling the human as a capable collaborator with three key aspects:

- **Human-centric egocentric view.** The assistant observes the world through the human's first-person view, which constantly shifts with body postures and hand movements.
- **Reliable execution of everyday actions.** Unlike robots, humans can reliably and naturally execute everyday actions (e.g., `GRAB`, `OPEN`) given high-level intents.
- **Autonomous translation of guidance into actions.** Humans can autonomously translate free-form guidance into executable actions based on what they see in the current scene.

Thus, while existing simulators provide a foundation, building a human-centric environment requires a new human representation, reliable semantic execution, and interaction mechanism.

Overall. To address these challenges, we propose InteractiveWorld, a human-centric interaction environment built on Isaac Sim (Fig. 2). From bottom to top, it consists of three layers: (1) The Physical Layer ensures stable, repeatable human actions that are physically consistent with the environment. (2) The Human Action Layer enables the "Human" to navigate and manipulate objects naturally and autonomously. (3) The Semantic Layer connects natural language guidance to executable human behaviors. Together, these layers make the Human ***an interactive agent that observes the environment, understands instructions, and executes actions independently***, rather than merely a directly controlled animated character. This enables InteractiveWorld to realistically simulate collaboration between human and assistant. We introduce each module in detail below.

2.1 PHYSICAL LAYER

The physical layer serves as a foundational 3D environment, providing realistic humans, assets, and physics engine. It supplies the basic physical states needed for high-level planning.

Stable Human Representation. We use SMPL+H (Romero et al., 2022) to model the full body and hands, placing a camera on the head for a first-person visual perspective. To prevent physical failures like random falls, the human model is driven kinematically. This ensures that the physical layer can successfully and deterministically execute any valid pose sequence provided by the upper layer.

Collision-Free Interaction. To prevent severe collisions between the human model and the environment, the system pre-checks navigation paths and arm reachability before execution. It automatically corrects minor collisions by smoothly adjusting the grasp position. For unsafe actions, the system rejects them and returns clear failure reasons to the upper layer.

Rich and scalable asset support. To support diverse daily tasks, we imported over 8,000 object and more than 50 environment assets from BEHAVIOR-1K, including furniture, containers, appliances, tools, and food. Thanks to Isaac Sim's asset import features, InteractiveWorld can load scanned models of real-world scenes and objects to simulate specific environments for future development.

2.2 HUMAN ACTION LAYER

Built on the physical foundation, the Human Action Layer enables the human model to perform high-level actions (e.g., grab, open) by dividing the execution into two stages:

Collision-Aware Auto-Navigation. Given a target object, the system first identifies a reachable standing position suitable for manipulation. It then constructs an obstacle map based on environment and plans a safe path using A* search (Hart et al., 1968). Following this path, the human executes continuous walking and turning motions, moving smoothly toward target without collisions.

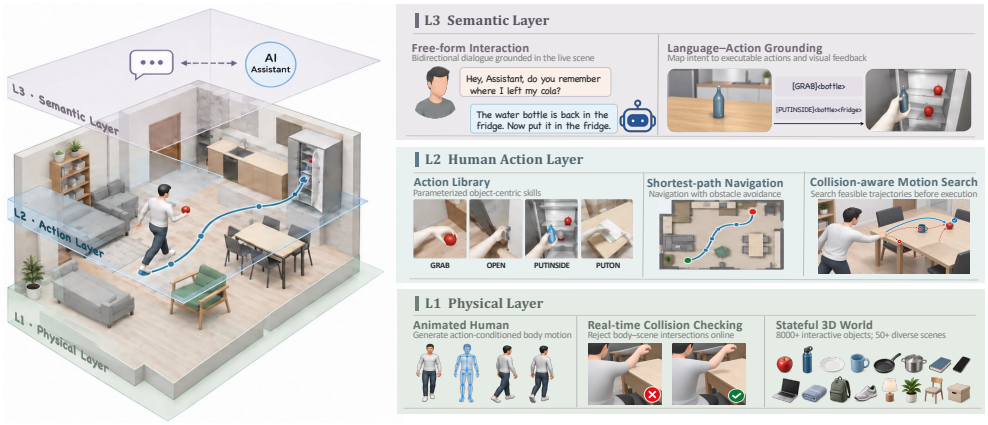


Figure 2: **InteractiveWorld Overview.** Three layers enable realistic human–assistant collaboration: (1) *Physical*: realistic humans, assets, and physics simulation; (2) *Human Action*: autonomous, stable high-level daily action execution; (3) *Semantic*: grounding free-form guidance into action sequences.

Target-Driven Adaptive Manipulation. Upon arrival, the system adapts its posture to stand, lean, or squat and plans a collision-free arm trajectory. Grasp position and hand posture depend on object shape, functional parts, and task. Predefined motion sequences for basic postures and actions let the hand wrap around a cup to lift it or grasp a handle to open a refrigerator. If the initial position offers no feasible trajectory to the object, the system automatically replans its position.

2.3 Semantic Layer

The Semantic layer builds upon the Human Action layer to link natural language with behavior, enabling the Assistant to guide the Human via dialogue rather than direct simulator control.

Unified action interface. We define 36 executable action primitives covering navigation, observation, manipulation, and daily activities. Supported actions range from basic (`[Walk]`, `[LookAt]`, `[Grab]`, `[Put]`, `[PutInside]`, `[Open]` and `[TurnOn]`) to complex operations (`[Pour]`, `[Cook]`, `[Cut]`). Commands specify only actions and targets (e.g., `[Grab]<cup>` or `[PutInside]<pizza> <oven>`) and are translated into goals for the human action layer, which autonomously handles underlying navigation, positioning, and body motions.

Verifiable multi-turn interaction. During interaction, the Assistant provides natural-language guidance based on the Human's first-person view and the conversation history. The Human interprets this guidance, then selects and executes the corresponding actions. After each action, the simulator updates the environment state and renders a new first-person view, forming a continuous "observe-guide-execute" loop. By tracking actual object positions and states throughout the interaction, the system can evaluates final object relationships to automatically verify task success.

By integrating these three layers, InteractiveWorld builds a comprehensive pipeline from the physical environment and human actions to natural language interaction. With an appropriate Human agent and Assistant agent, the system can simulates real-world scenarios where an egocentric AI Assistant assists a human, and provide verifiable outcomes to evaluate the Assistant's helpfulness.

3 InteractiveBench

Building on InteractiveWorld, we further propose InteractiveBench. To the best of our knowledge, this is the first closed-loop interactive benchmark designed for egocentric AI assistants.

Task Design. InteractiveBench contains 120 tasks across six categories (Tab. 1): Guidance, Planning, Accessibility, Decision, Prevention, and Finding. These cover the main everyday scenarios for egocentric AI assistance. All tasks undergo human review and meet three criteria: First, each addresses a realistic need for assistance, rather than a simple action the Human can complete unaided. Second, it requires the Assistant to use first-person information to observe, remember, reason, or plan,

Table 1: **Overview of the InteractiveBench Classification.** To simulate real-world scenarios, visual inputs are intentionally withheld from human agents during accessibility tasks.

Category	Evaluation focus	Example task
Guidance	**Guide user step by step** based on the user's progress, helping them complete task.	**Lab cleanup.** A new lab worker must put away 14 chemical bottles and clear broken glass. The assistant must use the labels, chemical knowledge and the safety rules to guide the cleanup.
Planning	**Outline the steps** for tasks within time and resource limits, flexibly adjust plan.	**Cooking for guests.** A chef need cook for guests with different food allergies, using one oven and pan within 30 minutes. The assistant must choose and schedule as many safe dishes as possible.
Accessibility	**Describe the scene for blind** or low-vision users so they can complete everyday tasks.	**Kitchen cleanup.** A blind user needs to clean a kitchen with broken glass, food, and dishes. The assistant must read the cleanup guide, describe the scene, and help the user finish safely.
Decision	**Help with daily decision** using current observations and assistant's knowledge.	**Supplier comparison.** A buyer sees an order for 40 filters, then sees two quotes with different box sizes in another room. The assistant must choose the cheaper offer for the full order.
Prevention	**Warn before mistakes happen**, while avoiding unnecessary interruptions.	**Leaving home.** A user is about to leave, but the fridge and living-room window were left open earlier. The assistant must remember these problems and remind the user before they leave.
Finding	**Help find missing objects** by remembering where they moved and guiding search.	**Finding stored tools.** A user cannot find the black and blue squeegees after storing them in a lidded box inside a cabinet in another room. The assistant must remember where they went.

and to help over multiple turns. Third, outcomes must be verifiable from clearly defined environment states. Appendix C details the full task construction and review process.

Task Evaluation. During evaluation, the target model serves as the Assistant agent and an auxiliary one acts as the Human agent. The Human agent is provided with all valid commands and can freely perform actions. However, it is instructed to strictly follow the Assistant's guidance rather than acting on its own intentions. Meanwhile, the Assistant agent is prompted to help the human complete the task. Detailed evaluation settings are provided in Appendix C. Overall, each evaluation consists:

- **Initial Scene.** Each task defines a complete starting world. This includes the scene layout, the human's starting position, and the locations and states of target and distractor objects.
- **Pre-interaction Setup (Warmup).** Before formal interaction begins, the human performs predefined actions and provides necessary opening context (e.g., looking around or reading). This stage establishes the background and gives assistant visual and temporal information.
- **Closed-loop Interaction.** After warmup, the assistant and human begin interaction. The assistant uses the egocentric view and conversation history to choose whether to stay silent, provide guidance, or end the task. The human then issues action commands and feedback based on the assistant's guidance and their own observations. The action executor performs these commands in the environment. After each action, both sides get a new chance to respond, and keyframes sampled during the action are added to the visual input.
- **Termination conditions.** The task ends and evaluation begins when the assistant outputs $done$. Alternatively, it force-terminates upon reaching a maximum limit of interaction.
- **Task Scoring.** Each task has a predefined set of core states (e.g., "the pizza is in the oven" or "the fridge is closed"). When the task ends, the system checks the actual states of the environment. The task is marked successful only if all required states are met.

Key advantages. InteractiveBench measures how the Assistant's advice affects subsequent actions and outcomes, offering three advantages: (1) *Long-horizon closed-loop interaction.* Each task requires multiple rounds of observation, guidance, action, and feedback. (2) *Multi-Skill integration.* Tasks require combining visual perception, memory, reasoning, and communication. (3) *Verifiable outcomes.* Success is determined by actual object states in the simulator. By simulating realistic everyday interactions, InteractiveBench directly measures whether the Assistant can help the Human.

Table 2: **Performance Across Six Task Categories.** Pass is the percentage of fully completed instances (all objectives met). MCR (Mean Completion Rate) is the average fraction of objectives completed per instance. We verified that, under our evaluation protocol, the same Human agent achieves a 100% pass rate with an Oracle Assistant and 0% without assistance.

Model	AVG.		Guidance		Planning		Accessibility		Decision		Prevention		Finding	
	Pass (%)	MCR (%)	Pass (%)	MCR (%)	Pass (%)	MCR (%)	Pass (%)	MCR (%)	Pass (%)	MCR (%)	Pass (%)	MCR (%)	Pass (%)	MCR (%)
Claude Fable 5.1	**41.7**	**54.8**	**20.0**	**32.3**	**25.0**	**45.6**	30.0	63.3	**85.0**	**95.1**	**50.0**	**50.0**	**40.0**	**42.7**
Claude Opus 5	**33.3**	**48.1**	15.0	**33.7**	**25.0**	**47.3**	25.0	56.5	**70.0**	86.4	30.0	30.0	**35.0**	35.0
Muse Spark 1.1	31.7	43.7	**20.0**	28.8	**25.0**	34.6	35.0	63.0	55.0	76.9	30.0	30.0	25.0	29.0
GPT-5.6 Sol	30.8	42.0	**20.0**	31.3	**25.0**	36.4	30.0	54.8	**70.0**	**89.4**	15.0	15.0	25.0	25.0
GPT-5.5	30.0	41.2	10.0	20.3	15.0	32.8	**45.0**	64.2	65.0	82.1	25.0	25.0	20.0	22.7
GLM 5.3 Flash	30.0	43.1	15.0	22.4	20.0	42.4	50.0	73.0	50.0	76.7	20.0	21.1	25.0	25.0
Claude Opus 4.8	28.3	41.7	0.0	15.6	20.0	41.0	30.0	49.5	50.0	72.4	**35.0**	**35.0**	**35.0**	**36.8**
Kimi K3	24.2	41.1	**20.0**	28.0	15.0	39.1	**40.0**	**65.8**	35.0	76.1	20.0	21.1	15.0	19.0
Qwen 3.8 Max	22.5	36.0	5.0	9.7	20.0	33.4	35.0	61.0	35.0	71.6	20.0	20.0	20.0	24.0
Gemini 3.1 Pro	20.8	37.8	10.0	22.7	10.0	37.5	35.0	**65.9**	50.0	79.1	10.0	10.0	10.0	11.8
GPT-5.6 Luna	20.0	37.2	5.0	22.0	5.0	28.7	20.0	51.3	40.0	66.2	30.0	30.0	20.0	24.9
DeepSeek V4.1 Flash	12.5	28.2	0.0	11.4	5.0	24.0	25.0	56.9	35.0	71.0	0.0	0.0	10.0	10.0

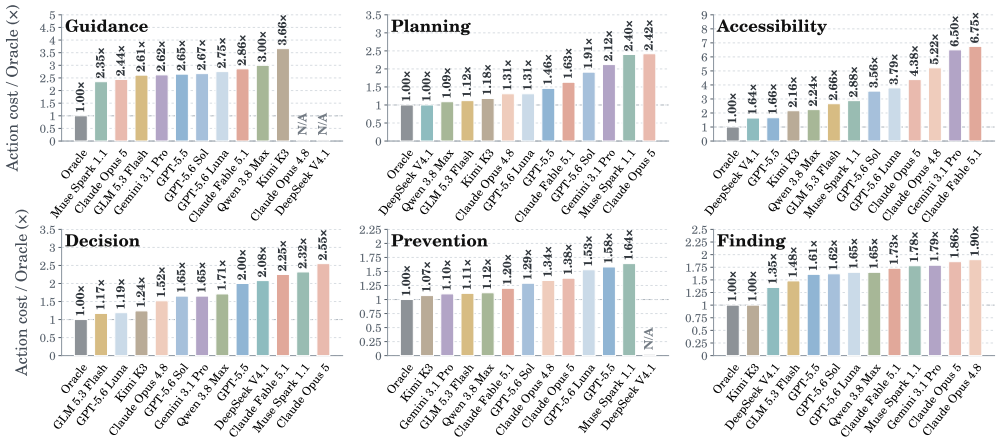


Figure 3: **Action Efficiency of Evaluated Models.** Efficiency is measured by the average ratio of a model's action steps to the task's oracle steps (with oracle assistant) on its individually passed tasks.

4 Evaluation

In this section, we evaluate the performance of frontier models as the Assistant agent on the proposed InteractiveBench to demonstrate the current challenge of MLLMs for real scenarios.

Models. We benchmark twelve state-of-the-art models as the assistant: Claude Fable 5.1, Claude Opus 5, and Claude Opus 4.8 from Anthropic (Anthropic, 2026a;c;b); GPT-5.6 Sol, GPT-5.6 Luna, and GPT-5.5 from OpenAI (OpenAI, 2026b;a); Muse Spark 1.1 from Meta (Meta AI, 2026); GLM-5.3-Flash from Z.ai (GLM-5-Team et al., 2026); Kimi K3 from Moonshot AI (Moonshot AI, 2026); Qwen3.8-Max from Alibaba (Alibaba Cloud, 2026); Gemini 3.1 Pro from Google (Google DeepMind, 2026); and DeepSeek-V4.1-Flash from DeepSeek (DeepSeek, 2026).

Evaluation Setup. To ensure consistent Human behavior and fair Assistant comparison, our main experiments keep Gemini 3.1 Pro as the Human agent and Section 4.2 ablates the impact of varying Human agents. Visual inputs are capped at 32 frames per turn, sampled at 4 frames per action. If this limit is exceeded, we retain the 8 most recent frames and uniformly sample 24 from the older history. Per turn, the Assistant outputs *interrupt* (with natural language instructions), *silent*, or *done* (when task is complete). The Human uses *action* for structured commands and *speak* for natural language replies. Detailed prompts and procedures are in the Appendix C.2.

Table 3: **Performance Across Nine Capabilities.** Each of the 120 tasks is annotated with required capabilities and evaluated using separate state-verifiable checks. For each capability, we report the percentage of passed evaluation points across all applicable tasks. Models are ordered by overall task pass rate in descending order. The nine capabilities are Object Recognition, Spatial Grounding, OCR, Long-horizon Memory, Planning, Knowledge, Error Detection, Proactivity, and Communication.

	Perception			Memory	Reasoning			Interaction		Overall
Model	**Objects**	**Spatial**	**OCR**	**Memory**	**Plan**	**Knowledge**	**Detect**	**Proactivity**	**Comm.**	**Avg.**
Claude Fable 5.1	63.0	61.2	62.5	**58.9**	**56.0**	54.9	51.9	44.4	79.6	**59.2**
Claude Opus 5	54.9	57.1	59.8	46.4	55.5	54.4	54.9	**53.4**	**80.1**	57.4
Muse Spark 1.1	**63.3**	**62.0**	64.3	48.9	48.1	54.2	45.5	33.5	76.5	55.1
GPT-5.6 Sol	53.3	54.1	48.7	49.2	49.6	43.2	51.1	40.2	78.5	52.0
GPT-5.5	52.8	54.1	47.7	40.2	44.7	45.7	46.6	34.0	78.0	49.3
GLM 5.3 Flash	59.3	56.7	**69.3**	49.9	54.6	**61.3**	**56.8**	45.9	69.8	58.2
Claude Opus 4.8	48.6	49.6	61.8	33.9	43.7	40.7	36.4	48.5	72.9	48.5
Kimi K3	59.1	51.8	63.8	50.1	45.9	59.7	53.9	41.9	74.8	55.7
Qwen 3.8 Max	43.8	43.5	47.2	39.6	39.5	47.1	32.2	32.1	62.6	43.1
Gemini 3.1 Pro	55.3	46.3	47.5	37.1	37.4	35.3	38.6	30.7	71.6	44.4
GPT-5.6 Luna	47.6	50.5	53.8	38.3	43.5	40.1	36.4	30.0	74.5	46.1
DeepSeek V4.1 Flash	57.4	46.0	51.1	31.1	35.7	40.9	33.3	21.9	65.9	42.6

Numbers: scores (%). **Bold**: best. Underline: second best.
Green: above the capability mean Red: below the capability mean All 12 models.

Deviation from capability mean (pp)
−18 −12 −6 0 +6 +12 +18

4.1 Benchmark Results and Analysis

We evaluate models on InteractiveBench through five lenses: (i) **Performance**: measuring the Human's task success rate and mean completion rate (Tab. 2); (ii) **Efficiency**: calculating the ratio of actual human actions to those required with an oracle assistant (Fig. 3); (iii) **Capability Assessment**: evaluating nine capabilities (perception, memory, reasoning, and interaction) based on predefined state conditions (Tab. 3); (iv) **Failure Analysis**: classifying the initial error in failed tasks into six categories (Fig. 4); and (v) **Complexity Impact**: analyzing the correlation between success rates and task length (Fig. 6). From these evaluations (details in Appendix D), we distill five key findings.

> **Finding 1:** Closed-loop assistance is far from solved: even current strongest MLLM cannot yet serve as a reliable egocentric AI Assistant, helping the Human complete fewer than half of the tasks.

As shown in Tab. 2, Claude Fable 5.1 achieves the highest average success rate at only 41.7%, with all other models at or below 33.3%. Since we verified that an Oracle Assistant achieves a 100% success rate with the same Human agent, this substantial gap indicates that current models still fail to provide reliable guidance, leaving immense room for improvement in collaborative AI.

> **Finding 2:** The difficulty lies in sustaining the loop: models make good one-shot decisions but break down when they must keep tracking progress and re-planning.

Performance varies significantly by task category. While best models reach an 85.0% success rate on Decision tasks, they struggle heavily on Guidance (20.0%) and Planning (25.0%). This gap stems from task complexity: Decision tasks often rely on a single scene judgment, whereas Guidance and Planning demand continuous tracking, anomaly detection, and dynamic replanning. Consistent with this, Fig. 6 shows that model performance drops quickly as tasks get longer, especially on guidance and planning. Together, these results show what existing static evaluations fail to capture.

> **Finding 3:** Most failures stem from perception and memory: models misread scenes or lose track of changes, revealing current models' poor foundational visual capabilities for interaction.

Across all models, visual perception and memory errors account for 70.4%–85.7% of first errors (Fig. 4). Because closed-loop interaction requires continuously tracking dynamic scenes, a single misperception or forgotten state cascades into all subsequent instructions. This result indicates that current models' foundational capabilities for interaction remain insufficient. Furthermore, the

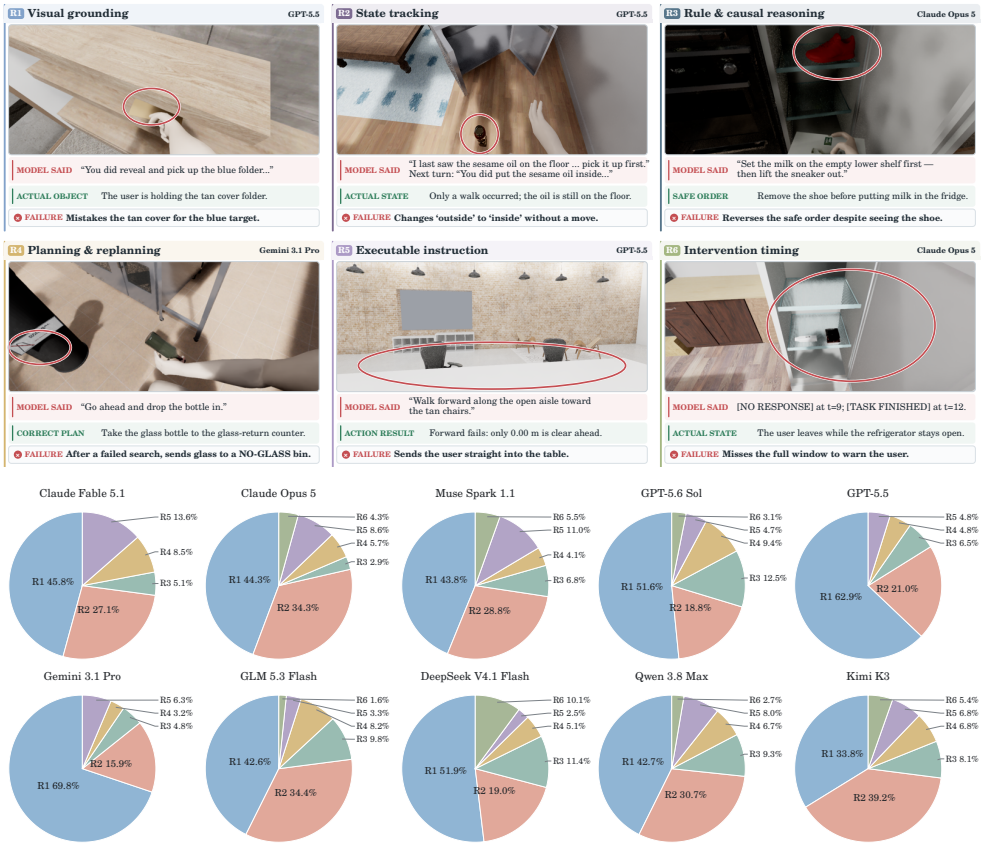


Figure 4: **Examples and Distribution of Error Types.** We analyze initial failure of failed tasks and grouped into six types. (Top) Examples of each error type. (Bottom) Error type distributions.

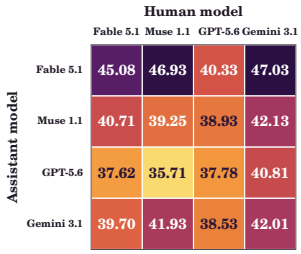
breakdown also varies across models: visual perception accounts for 69.8% of Gemini 3.1 Pro's first errors, whereas Kimi K3 fails more often on memory than on perception (39.2% vs. 33.8%).

> **Finding 4:** Communication is critical: if the Assistant cannot express its plan as clear, step-by-step guidance, strong performance in other capabilities does little to help the Human.

Models with higher success rates generally score higher on most capabilities (Tab. 3), but the average does not determine success. GLM-5.3-Flash and Claude Fable 5.1 have nearly identical average capability scores (58.2% vs. 59.2%), yet GLM-5.3-Flash scores markedly lower on communication (69.8% vs. 79.6%) and completes 11.7 points fewer tasks (30.0% vs. 41.7%). The error analysis points the same way: even for Claude Fable 5.1, 13.6% of first errors arise when the Assistant fails to turn the task goal into guidance the Human can follow (R5 example in Fig. 4). Closed-loop success is thus bounded by the weakest capability in the chain rather than by the average, and communication of this kind is not measured by static question-answering benchmarks before.

> **Finding 5:** Action efficiency also remains low: current models lead the Human through many unnecessary actions, making efficiency an important dimension for improving assistance alongside task success.

Compared to the Oracle, almost all models exhibit severe action redundancy (Fig. 3), peaking in Accessibility and Guidance (where models require $2.35\times$–$3.66\times$ the Oracle's cost). Furthermore, efficiency is highly task-dependent rather than a fixed model trait. For instance, Claude Fable 5.1, despite being the most successful overall, is the least efficient on Accessibility ($6.75\times$), yet highly efficient on Prevention ($1.20\times$). Similarly, Kimi K3 matches the Oracle on Finding ($1.00\times$) but suffers the highest cost on Guidance ($3.66\times$). Since unnecessary instructions directly translate to wasted human effort, optimizing interaction efficiency is as critical as maximizing success rates.



Model	Guide	Plan	Access.	Decide	Preven.	Find
Fable 5.1	20.0%	25.0%	30.0%	85.0%	50.0%	40.0%
+ Memory	15.0%	20.0%	30.0%	90.0%	55.0%	50.0%
GPT-5.6-sol	20.0%	25.0%	30.0%	70.0%	15.0%	25.0%
+ Memory	20.0%	20.0%	15.0%	65.0%	30.0%	40.0%
Gemini 3.1 Pro	10.0%	10.0%	35.0%	50.0%	10.0%	10.0%
+ Memory	15.0%	10.0%	45.0%	55.0%	15.0%	20.0%

Figure 5: (Left) **Human Agent Impact.** We report average MCR across Guidance, Planning, and Accessibility (tasks requiring maximal human involvement) when cross-evaluating four diverse models interchangeably as assistant and human agents. (Right) **Memory Augmentation Impact.** We report model pass rates with vs. without inference-time memory augmentation.

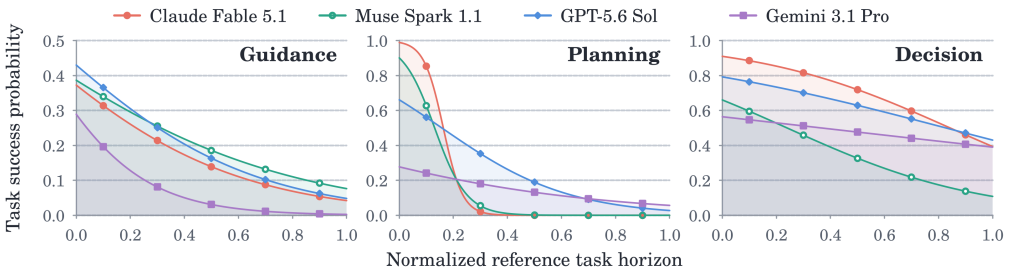


Figure 6: **Performance Across Task Horizons.** We report model success rates against task lengths (derived from oracle actions). Task horizons are normalized per category on the x-axis (0: shortest, 1: longest), while the y-axis shows the average pass rate at each normalized length step.

4.2 Ablation Study

Furthermore, we conduct ablation studies to investigate (1) the effectiveness of inference-time memory augmentation, and (2) the impact of varying the Human agent model on task performance.

Inference-time memory augmentation. To address state-tracking limitations, we append a hidden `$Memory$` field to the Assistant's responses, allowing it to record events for subsequent turns. This augmentation improves success rates by 5–15% on Prevention and Finding across all three models (Fig. 5). However, its effects on Decision, Guidance, and Accessibility vary across models, while Planning success drops by 5% for two models. These mixed results suggest that this simple memory augmentation alone is insufficient for reliable long-horizon assistance, motivating future work on training native memory mechanisms for dynamic interaction. See Appendix D for details.

Effect of the Human agent. To evaluate the Human's impact, we test all 16 Assistant–Human pairings among four diverse models, reporting MCR on Guidance, Planning, and Accessibility, long-horizon tasks reliant on instruction following (Fig. 5). An Assistant's performance varies with its Human partner, although rankings remain relatively stable: Fable 5.1 consistently ranks first, while GPT-5.6 ranks last. Performance also differs across roles: although Gemini 3.1 ranks second or third as an Assistant, all four Assistants achieve their highest observed MCR when it acts as the Human. This implies that the challenge in Human agent is different from the Assistant agent and Gemini can be a good proxy of Human for our benchmark. Future work will explore behavioral calibration to better align simulated Human agents with real users.

5 Conclusion

We introduced InteractiveWorld, a verifiable environment for closed-loop human-assistant interaction, and InteractiveBench, a dynamic benchmark of 120 multi-turn tasks. InteractiveWorld measures interactive abilities with verifiable rewards, while InteractiveBench evaluates how models assist people in real world. They provide a foundation for developing and evaluating interactive MLLMs.

AI USE STATEMENT

In this work, we used generative AI tools to generate candidate task specifications and clean/reformat task data during the construction of InteractiveBench (Appendix C.1); every AI-generated task was manually checked by the authors and validated by the state-based evaluators. We did not use generative AI tools to propose hypotheses, design the methodology or experiments, implement the simulator, or interpret results; translation and proofs are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript for readability. The MLLMs evaluated in Section 4 are the subject of study rather than writing or research aids. We have reviewed all AI-assisted work and take full responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Alibaba Cloud. Alibaba unveils qwen3.8-max: Its largest and most capable flagship model to date. https://www.alibabacloud.com/en/press-room/alibaba-unveils-qwen3-8-max, August 2026. Accessed: 2026-09-21.

Anthropic. Claude fable 5.1 and claude mythos 5.1. https://www.anthropic.com/claude-fable-and-mythos-5-1, September 2026a. Accessed: 2026-09-21.

Anthropic. Introducing Claude Opus 4.8. https://www.anthropic.com/news/claude-opus-4-8, 2026b.

Anthropic. Introducing claude opus 5. https://www.anthropic.com/news/claude-opus-5, July 2026c. Accessed: 2026-09-21.

Keshigeyan Chandrasegaran, Agrim Gupta, Lea M. Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. Hourvideo: 1-hour video-language understanding. 2024.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, Siddharth Patki, Ishita Prasad, Xavi Puig, Akshara Rai, Ram Ramrakhya, Daniel Tran, Joanne Truong, John M. Turner, Eric Undersander, and Tsung-Yen Yang. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. 2025.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. 2024.

Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Rui-Lan Xu, Ying Shan, and Xihui Liu. Egoplan-bench: Benchmarking multimodal large language models for human-level planning. International Journal of Computer Vision, 2023.

S Cheng, Z Guo, J Wu, K Fang, P Li, H Liu, and Y Liu. Egothink: Evaluating first-person perspective thinking capability of vision-language models. 2024a.

Sijie Cheng, Kechen Fang, Yangyang Yu, Sicheng Zhou, Bohao Li, Ye Tian, Tingguang Li, Lei Han, and Yang Liu. Videgothink: Assessing egocentric video understanding capabilities for embodied ai. arXiv (Cornell University), 2024b.

D. Damen, Hazel Doughty, G. Farinella, S. Fidler, Antonino Furnari, E. Kazakos, D. Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. 2018.

Avijit Dasgupta, Shayon Dasgupta, Zakaria Laskar, C. V. Jawahar, and Karteek Alahari. Puzzlemate: Benchmarking mllms for egocentric puzzle assistance. ICPR (16), 2026.

DeepSeek. Introducing deepseek-v4.1-flash: Smarter, faster, more efficient. https://www.deepseek.com/en/news/deepseek-v4-1-flash/, September 2026. Accessed: 2026-09-21.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S. Sukhatme. Dialfred: Dialogue-enabled agents for embodied instruction following. IEEE Robotics and Automation Letters, 2022.

GLM-5-Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Google DeepMind. Gemini 3.1 pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, February 2026. Accessed: 2026-09-21.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran K. Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. 2022.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zachary Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, María Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Dutt 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 González, 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 K. 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 Arbeláez, Gedas Bertasius, Dima Damen, Jakob J. Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard A. Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shout, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. 2024.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern., 1968.

Yifei Huang, Jilan Xu, Baoqi Pei, Yuping He, Guo Chen, Lijin Yang, Xinyuan Chen, Yaohui Wang, Zheng Nie, Jinyao Liu, Guoshun Fan, Dechen Lin, Fang Fang, Kunpeng Li, Chang Yuan, Yali Wang, Yu Qiao, and Limin Wang. Vinci: A real-time embodied smart assistant based on egocentric vision-language model, 2024. URL https://arxiv.org/abs/2412.21080.

Eric Kolve, Roozbeh Mottaghi, Abhinav Gupta, and A. Gupta. Ai2-thor: An interactive 3d environment for visual ai. arXiv.org, 2017.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, C. Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. 2021.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1, 000 everyday activities and realistic simulation. 2022.

Siyao Li, Jiawei Gu, Shuai Liu, Kairui Hu, Zekun Li, Linjie Li, Chengcheng Tang, Po-Chen Wu, Ivan Shugurov, Lingni Ma, Michael Zollhoefer, Sizhe An, Abhay Mittal, Amy Zhao, Ranjay Krishna, Manling Li, Ziwei Liu, and Chuan Guo. Humanclaw: Can vision-language models act through a body? arXiv (Cornell University), 2026.

Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, Hongfa Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. 2022.

Chang Liu, Jiawei Zhang, Tao Zhang, Ye Wang, Hongyu Zhou, and Qin Jin. Hribench: Benchmarking interaction-centric human-robot collaboration. arXiv (Cornell University), 2026a.

Yunqi Liu, Tong Niu, Zitong Wang, Zhenlong Dai, Yuqi Qing, Weiqiang Wang, and Jian Liu. Egobench: An interactive egocentric multimodal benchmark for tool-using agents. arXiv (Cornell University), 2026b.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. 2023.

Meta AI. Introducing muse spark 1.1. https://research.meta.ai/blog/introducing-muse-spark-meta-model-api, July 2026. Accessed: 2026-09-21.

Nvidia Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Munoz, Xinjie Yao, Ren'e Zurbrugg, N. Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, M. Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, H. Mazhar, M. Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander

Poddubny, Nathan D. Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, R. Thaker, Matthew Trepte, Karl Van Wyk, Fang Yu, Alexander Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, A. V. Kuruttukulam, Michael A. Lin, A. Mandlekar, Karsten Patzwaldt, John Welsh, Hui Zhao, Fatima Anes, Jean-Francois Lafleche, Nicolas Moenne-Loccoz, Soowan Park, Rob Stepinski, D. Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, P. D. H. Ciechomski, Gilles Daviet, M. Mohajerani, Julia von Muralt, V. Reutskyy, Michael Sauter, S. Schirm, Eric L. Shi, Pierre Terdiman, K. Vilella, Tobias Widmer, Gordon Yeoman, Tiffany Chen, Sergey Grizan, Cathy Li, Lotus Li, Connor Smith, Rafael Wiltz, Kostas Alexis, Yan Chang, David Chu, LinxiJimFan, Farbod Farshidian, Ankur Handa, Spencer Huang, Marco Hutter, Yashraj S. Narang, Soha Pouya, Shiwei Sheng, Yuke Zhu, Miles Macklin, Ádám Moravánszky, Philipp Reist, Yunrong Guo, David Hoeller, and Gavriel State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. arXiv preprint arXiv:2511.04831, 2025.

Moonshot AI. Kimi k3: Open frontier intelligence. https://www.kimi.ai/blog/kimi-k3, July 2026. Accessed: 2026-09-21.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. 2024.

NVIDIA. NVIDIA Isaac Sim. https://developer.nvidia.com/isaac/sim, 2025.

OpenAI. GPT-5.5 system card. https://openai.com/index/gpt-5-5-system-card/, 2026a.

OpenAI. Gpt-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026b. Accessed: 2026-09-21.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gökhan Tür, and Dilek Hakkani-Tür. Teach: Task-driven embodied agents that chat. 2022.

Dhruvesh Patel, Hamid Eghbalzadeh, Nitin Kamra, Michael Louis Iuzzolino, Unnat Jain, and Ruta Desai. Pretrained language models as visual planners for human assistance. 2023.

Yuhang Peng, Yizhou Pan, Xinning He, Jihaoyu Yang, Xinyu Yin, Han Wang, Xiaoji Zheng, Chao Gao, and Jiangtao Gong. Freeaskworld: An interactive and closed-loop simulator for human-centric embodied ai. 2026.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. 2018.

Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B. Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. 2022.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Théophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots. 2023.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction, 2025. URL https://arxiv.org/abs/2501.03218.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. CoRR, 2022.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. 2019.

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. 2020.

Alessandro Suglia, Claudio Greco, Katie Baker, Jose L. Part, Ioannis Papaioannou, Arash Eshghi, Ioannis Konstas, and Oliver Lemon. Alanavlm: A multimodal embodied ai foundation model for egocentric video understanding. 2024.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John M. Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel X. Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. 2022.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Rajesh, Yong Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating gpu parallelized robot simulation and rendering for generalizable embodied ai with maniskill3. 2025.

Jesse Thomason, Michael Murray, M. Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. 2019.

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Chain-of-tool-thought for ultra-long egocentric video reasoning. arXiv (Cornell University), 2025.

Mrinal Verghese, Brian Chen, Hamid Eghbalzadeh, Tushar Nagarajan, and Ruta Desai. User-in-the-loop evaluation of multimodal llms for activity assistance. 2025.

Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, Peizhou Cao, Wenye Yu, Zichao Ye, Jialun Li, Junfeng Long, Zirui Wang, Huiling Wang, Ying Zhao, Zhongying Tu, Yu Qiao, Dahua Lin, and Jiangmiao Pang. Grutopia: Dream general robots in a city at scale. arXiv (Cornell University), 2024.

X Wang, T Kwon, M Rad, B Pan, I Chakraborty, S Andrist, D Bohus, A Feniello, and B Tekin. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. 2023a.

Ying Wang, Yanlai Yang, and Mengye Ren. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos. arXiv (Cornell University), 2023b.

Junbin Xiao, Nanxin Huang, Hao Qiu, Zhulin Tao, Xun Yang, Richang Hong, Meng Wang, and Angela Yao. Egoblind: Towards egocentric visual assistance for the blind. 2025.

Jiaqi Yan, Ruilong Ren, Jingren Liu, Shuning Xu, Ling Wang, Yiheng Wang, Xinlin Zhong, Yun Wang, Long Zhang, Xiangyu Chen, Changzhi Sun, Jixiang Luo, Dell Zhang, Hao Sun, Chi Zhang, and Xuelong Li. Teleego: Benchmarking egocentric ai assistants in the wild. ArXiv.org, 2025.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, Bei Ouyang, Zhengyu Lin, Marco Cominelli, Zhongang Cai, Bo Li, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, Lei Yang, and Ziwei Liu. Egolife: Towards egocentric life assistant. 2025a.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. 2025b.

Sicheng Yang, Yukai Huang, Weitong Cai, Shitong Sun, Fengyi Fang, You He, Yiqiao Xie, Jiankang Deng, Hang Zhang, Jifei Song, and Zhensong Zhang. Egocentric co-pilot: Web-native smart-glasses agents for assistive egocentric ai. 2026.

Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, Jiasen Lu, and Yinfei Yang. Mm-ego: Towards building egocentric multimodal llms for video qa. 2025.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. 2023.

Y Zhang, XL Dong, Z Lin, A Madotto, A Kumar, B Damavandi, J Chai, and S Moon. Proactive assistant dialogue generation from streaming egocentric videos. 2025a.

Yulin Zhang, Cheng Shi, Yang Wang, and Sibei Yang. Eyes wide open: Ego proactive video-llm for streaming video. 2025b.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. 2023.

Qinhong Zhou, Hongxin Zhang, Xiangye Lin, Zheyuan Zhang, Yutian Chen, Wenjun Liu, Zunzhe Zhang, Sunli Chen, Lixing Fang, Qiushi Lyu, Xinyu Sun, Jincheng Yang, Zeyuan Wang, Bao Chi Dang, Zhehuan Chen, Daksha Ladia, Jiageng Liu, and Chuang Gan. Virtual community: An open world for humans, robots, and society. 2025.

APPENDIX CONTENTS

A Background & Related Work 16
B Implementation Details of InteractiveWorld 18
B.1 Implementation Details of Physical Layer 18
B.2 Implementation Details of Human Action Layer 20
B.3 Implementation Details of Semantic Layer 21
C Implementation Details of InteractiveBench 25
C.1 Benchmark Construction 25
C.2 Evaluation Protocol 27
C.3 Evaluation Metrics 30
C.4 Task Catalog 31
D Evaluation Setting Details and More Experiment Results 36
D.1 Experiment Setting Detail of Error Type Analysis 36
D.2 Experiment Setting Detail of Memory Augmentation 36
D.3 Additional Experiments about Comparative of Dynamic vs. Static 38
D.4 Additional Experiment of Task Horizon and Model Performance 39
D.5 Additional Human-Agent Validation with Real Human Participants 40
E Future Work 42

A BACKGROUND & RELATED WORK

Egocentric AI Assistant Benchmarks. Large-scale first-person datasets such as EPIC-KITCHENS Damen et al. (2018), Ego4D Grauman et al. (2022), and Ego-Exo4D Grauman et al. (2024) have laid the foundation for egocentric perception. Building on them, a growing body of benchmarks examines whether MLLMs can act as egocentric assistants: EgoSchema Mangalam et al. (2023) and HourVideo Chandrasegaran et al. (2024) test long-form video understanding; EgoThink Cheng et al. (2024a) and VidEgoThink Cheng et al. (2024b) evaluate first-person reasoning; EgoPlan-Bench Chen et al. (2023) targets next-step planning; EgoLife Yang et al. (2025a) and TeleEgo Yan et al. (2025) probe long-term memory over continuously recorded daily life; and EgoBlind Xiao et al. (2025) focuses on visual assistance for blind users. Closer to task guidance, HoloAssist Wang et al. (2023a) records real-world instructor–performer collaborations, while ProAssist Zhang et al. (2025a) and the ESTP task Zhang et al. (2025b) evaluate when and what a proactive assistant should say along a streaming egocentric video. Despite their diversity, these benchmarks are open-loop: model outputs are scored against fixed recordings and never influence what happens next. EgoBench Liu et al. (2026b) introduces an LLM-simulated user and tool calls, yet its visual context remains pre-recorded. User-in-the-loop studies Verghese et al. (2025); Dasgupta et al. (2026) do close the loop with real participants, and Verghese et al. Verghese et al. (2025) further show that offline metrics do not reflect online performance; however, such studies involve only a handful of participants and are costly to scale and hard to reproduce. InteractiveBench closes this loop in simulation: every instruction changes the human's actions and thus the world, and success is verified automatically from final object states.

Improving Egocentric AI Assistants. In parallel, many works improve the models and frameworks behind egocentric assistants. Early efforts learn egocentric video–language representations from narrated first-person videos Lin et al. (2022); Zhao et al. (2023), and subsequent MLLMs such as MM-Ego Ye et al. (2025), AlanaVLM Suglia et al. (2024), and EgoGPT Yang et al. (2025a) are tuned on large-scale egocentric instruction data. To handle continuous input, VideoLLM-online Chen et al. (2024), Dispider Qian et al. (2025), and proactive assistant models Zhang et al. (2025a;b) learn to decide when to respond as the video unfolds. For long-horizon memory, LifelongMemory Wang et al.

Table 4: **Comparison with existing egocentric AI assistant benchmarks and embodied simulators.** InteractiveWorld is the only environment that satisfies all six properties. Ego. View: first-person observations of a human. Human Actor: the task is physically performed by a (real or simulated) human, not a robot. Guided Human: the human autonomously grounds the agent's free-form language guidance into actions. NL Dialog: multi-turn natural-language interaction. Closed Loop: agent outputs change the environment and the subsequent observations. Verif. Success: success is checked against ground-truth environment states rather than by answer matching or LLM judges. ◐: partially supported.

Method	Primary Focus	Platform	Agent Output	Human-Centric			Interaction & Eval.		
				Ego. View	Human Actor	Guided Human	NL Dialog	Closed Loop	Verif. Success
Egocentric AI assistant benchmarks (offline, pre-recorded real-world videos)									
EgoSchema Mangalam et al. (2023)	Long-form video QA	Ego4D	MCQ answer	✓	✓	✗	✗	✗	✗
EgoThink Cheng et al. (2024a)	First-person VQA	Ego4D	Free-form answer	✓	✓	✗	✗	✗	✗
EgoPlan-Bench Chen et al. (2023)	Egocentric planning	Ego4D, EPIC	MCQ (next step)	✓	✓	✗	✗	✗	✗
EgoLife Yang et al. (2025a)	Long-term life QA	Aria glasses	MCQ answer	✓	✓	✗	✗	✗	✗
HoloAssist Wang et al. (2023a)	Task guidance	HoloLens 2	Labels / poses	✓	✓	◐	◐	✗	✗
ProAssist Zhang et al. (2025a)	Proactive guidance	6 datasets	Dialogue turns	✓	✓	✗	✓	✗	✗
Robot-centric embodied simulators									
Habitat 2.0 Szot et al. (2022)	Rearrangement	Habitat	Robot actions	✗	✗	✗	✗	✓	✓
BEHAVIOR-1K Li et al. (2022)	Household activities	OmniGibson	Robot actions	✗	✗	✗	✗	✓	✓
ManiSkill3 Tao et al. (2025)	Manipulation	SAPIEN	Robot actions	✗	✗	✗	✗	✓	✓
Isaac Lab Mittal et al. (2025)	Robot learning	Isaac Sim	Robot actions	✗	✗	✗	✗	✓	✓
Human- and interaction-oriented simulators									
VirtualHome Puig et al. (2018)	Activity programs	Unity	Action scripts	✓	✓	✗	✗	✓	✓
TEACh Padmakumar et al. (2022)	Dialogue-guided tasks	AI2-THOR	Follower actions	◐	✗	◐	✓	✓	✓
Habitat 3.0 Puig et al. (2023)	Human–robot collab.	Habitat	Robot actions	✗	◐	✗	✗	✓	✓
GRUtopia Wang et al. (2024)	City-scale society	Isaac Sim	Robot actions	✗	✗	✗	✓	✓	✓
FreeAskWorld Peng et al. (2026)	Direction inquiry	Unity	Nav. + queries	✗	✗	✗	✓	✓	✓
HumanCLAW Li et al. (2026)	Acting via a body	Habitat	Body skills	✓	✓	✗	✗	✓	✓
InteractiveWorld (Ours)	**Human–assistant collab.**	**Isaac Sim**	**NL guidance**	✓	✓	✓	✓	✓	✓

(2023b) and EgoRAG Yang et al. (2025a) retrieve from textualized histories, and Ego-R1 Tian et al. (2025) trains an agent with reinforcement learning to invoke retrieval and perception tools step by step over week-long videos. At the system level, VLaMP Patel et al. (2023) leverages pretrained language models to plan the next steps for human assistance, while Vinci Huang et al. (2024) and Egocentric Co-Pilot Yang et al. (2026) integrate memory, planning, and tool use into assistants deployable on portable devices. However, these methods are trained and evaluated on static data or small-scale user studies, so it remains unclear whether their gains carry over to closed-loop assistance, where the assistant must track how its own guidance changes the scene and the human. InteractiveWorld offers a verifiable testbed to answer this question, and its state-based rewards can further support training interactive assistants.

Robot-Centric Embodied Simulators. Simulators have long provided scalable and reproducible testbeds for embodied AI. AI2-THOR Kolve et al. (2017), Habitat Savva et al. (2019); Szot et al. (2022), iGibson Li et al. (2021), and BEHAVIOR-1K Li et al. (2022) offer interactive household scenes for navigation, rearrangement, and everyday activities; ManiSkill3 Tao et al. (2025) and Isaac Lab Mittal et al. (2025) provide GPU-parallel physics and rendering for large-scale robot learning; and RoboCasa Nasiriany et al. (2024) and GRUtopia Wang et al. (2024) scale up scene diversity from kitchens to entire cities. On top of these platforms, benchmarks such as ALFRED Shridhar et al. (2020) and EmbodiedBench Yang et al. (2025b) evaluate language-conditioned agents and MLLMs by having them directly command a robot. In these settings, the evaluated model is the actor: it outputs actions that the simulator executes on a robot body, while humans are either absent or merely part of the scene. HumanCLAW Li et al. (2026) replaces the robot with a physically simulated human body and decouples decision making from motor control, yet the VLM still decides every move of that body. In contrast, the assistant in InteractiveWorld never acts on the world directly; it can influence the outcome only through natural-language guidance that an autonomous human must interpret and carry out, mirroring how assistants help people in practice.

Interaction-Oriented Simulators. Another line of work brings humans or dialogue into simulation. In dialog-based embodied tasks such as CVDN Thomason et al. (2019), TEACh Padmakumar et al. (2022), and DialFRED Gao et al. (2022), an instruction giver guides a follower through natural

language; however, models are typically evaluated as the follower, with guidance replayed from recorded human dialogues or produced by a rule-based oracle. Human–AI collaboration platforms instead simulate human-like agents alongside AI helpers: VirtualHome Puig et al. (2018) and Watch-And-Help Puig et al. (2022) model household activities of humanoid avatars; Habitat 3.0 Puig et al. (2023) and PARTNR Chang et al. (2025) study human–robot coordination with simulated humanoids and real humans in the loop; and CoELA Zhang et al. (2023), Virtual Community Zhou et al. (2025), and HRIBench Liu et al. (2026a) extend this to LLM-driven cooperation, open-world societies, and role-conditioned interaction. FreeAskWorld Peng et al. (2026) further employs LLM-driven avatars that give directions to a navigating agent. Across these platforms, the AI remains the physical actor, while humans serve as scripted partners, co-workers, or sources of information. InteractiveWorld reverses these roles: the AI is a disembodied assistant that shares the human's view, and an autonomous MLLM-driven human decides how to act on its guidance, enabling closed-loop, verifiable evaluation of assistants that help people rather than act for them.

B Implementation Details of InteractiveWorld

As mentioned in the main text, the simulator comprises three layers: a *physical layer* that maintains a geometrically consistent human, scene, and set of rigid objects (§B.1); a *human-action layer* that converts semantic goals into locomotion and manipulation trajectories (§B.2); and a *semantic layer* that exposes these capabilities through a compact command language and a verifiable multi-turn interaction loop (§B.3).

B.1 Implementation Details of Physical Layer

B.1.1 Stable Human Representation

The camera wearer is a fixed-shape SMPL+H model: the male model with zero shape coefficients ($\boldsymbol{\beta} = \mathbf{0}$), a 52-joint kinematic tree, 6,890 vertices, and 13,776 triangles. Each frame is parameterized by a 156-D axis–angle vector

$$\mathbf{q} = [\boldsymbol{\omega}_{\text{root}}, \boldsymbol{\theta}_{\text{body}}, \boldsymbol{\theta}_{\text{left}}, \boldsymbol{\theta}_{\text{right}}] \in \mathbb{R}^{3+63+45+45}, \tag{1}$$

comprising the root orientation, 21 body joints (torso, head, arms, legs), and 15 articulated joints per hand; root translation is stored separately in world coordinates. A fixed 90° rotation about x maps the y-up model to the z-up world. Standard SMPL+H forward kinematics and linear blend skinning yield joint transforms and deformed vertices. The body mesh is authored once at `/World/Body`; each frame updates only vertex positions and normals, so topology, material binding, and camera attachment remain fixed.

Locomotion reuses body motion from ACCAD/AMASS SMPL-X clips: shared body rotations are copied, incompatible shape parameters are discarded, and the hand channels either keep a task-specific pose or are overwritten by the manipulation controller. The importer detects the source up axis, canonicalizes the initial heading, and estimates the gait period by autocorrelation. Gait phase advances with distance traveled rather than wall-clock time, so slowing, turning in place, or pausing to replan causes no foot sliding; the lowest foot is continuously aligned with the estimated support floor. Manipulation poses are layered onto the same skeleton, avoiding a model switch between walking, reaching, and object interaction.

The human is deliberately kinematic rather than a ragdoll, which makes the camera trajectory reproducible and prevents falling or numerical drift under contact. Collisions are nevertheless enforced: every desired pose is screened against an collision representation before it is committed.

B.1.2 Collision-Free Interaction

Proxy geometry. Rendering uses the full SMPL+H surface, whereas collision queries use a compact proxy rig of 17 capsules (pelvis, spine, neck/collars, upper and lower arms, hands, thighs, shins, feet) plus a head sphere. Capsule endpoints are regenerated from the current joints and radii are derived from the corresponding body segments; hand capsules extend from the wrist toward the palm. Contacts between adjacent segments are filtered, while contacts with the scene, manipulated objects, and non-adjacent body parts remain active.

Shared collision world. A single collision world serves scene initialization, navigation, stance selection, reaching, and post-action validation. It enumerates collision-enabled scene geometry, follows USD instance proxies, preserves each mesh's vertical extent, and groups multi-mesh articulated links under one owning object. Proposed poses are checked without one-frame lag by sampling spheres along each capsule at the *requested* pose rather than the last rendered one. Candidate stances are tested both in the idle configuration and with the relevant arm extended; the intended contact region around the target (nominal radius $0.22\,\mathrm{m}$) may be excluded from the reach test, but the rest of the target and the scene remain obstacles. On collision, bisection estimates the penetration and the controller tries another stance, posture, or approach. Manipulation trajectories use a stricter acceptance tolerance of $\approx 2\,\mathrm{mm}$.

Carried objects. The held object's collision geometry is propagated to every sampled wrist pose and tested against the scene and all non-held objects; only the intended hand–object grasp contact and the commanded target contact are allow-listed. An arm-safe trajectory is therefore still rejected if the carried object would strike a shelf, container rim, or neighboring object. Placement candidates are checked against non-target collision owners before release, and a post-settle overlap check catches contacts introduced by rigid-body dynamics.

Object physics. Static architectural and furniture assets use their authored collision meshes when available. Movable BEHAVIOR objects use the asset's authored convex pieces or, failing that, a convex decomposition of the concave visual mesh rather than a single concave dynamic collider. Initial packing enforces a nominal $2\,\mathrm{cm}$ separation; objects are then settled under gravity ($9.81\,\mathrm{m\,s^{-2}}$, 60 physics steps/s) until their speed falls below $10^{-3}\,\mathrm{m\,s^{-1}}$. A grasped object becomes kinematic and follows the wrist through a fixed relative transform. On release it becomes dynamic, is settled and, if necessary, minimally depenetrated, and its resulting transform is written back to the semantic state. Contents of a carried container follow via their stored relative transforms.

Scope of the guarantee. "Collision-free" denotes an operational guarantee: every planned route and manipulation is screened against the same geometry and rejected or repaired if it exceeds the configured contact tolerance. It does not claim exact separation between continuous triangle surfaces. In the egocentric evaluation configuration, small body-only contacts may be retained as diagnostic telemetry provided the camera/eye volume stays clear, whereas an object penetration above $5\,\mathrm{cm}$ fails the final placement gate. This prevents harmless skin-level numerical contacts from destabilizing behavior while still rejecting visually or physically consequential interpenetration.

B.1.3 Rich and Scalable Asset Support

We use BEHAVIOR-1K v3.9.0, which contains 1,829 object categories, 8,635 model directories, and 51 scene layouts. The layouts place 13,516 object instances covering 3,707 distinct category/model pairs. The generic importer omits purely decorative or non-interactive classes (e.g., ceilings, pictures, mirrors, windows, curtains, lighting fixtures), leaving 11,428 instances and 3,189 distinct models, all present in the deployed cache. At the time of this audit, the cache holds 6,831 plain-USD models spanning all 1,829 categories. Of the 51 layouts, 44 pass generic task admission (successful bake, usable floors and walls, and at least one work surface or container). BEHAVIOR's 996 material/particle systems are catalogued separately and are not counted as manipulable rigid objects in our action interface.

The layouts span domestic, office, retail, educational, hospitality, and leisure environments, e.g., `Beechwood_0_int`, `Ihlen_0_int`, `house_single_floor`, `office_cubicles_right`, `restaurant_diner`, `school_chemistry`, `grocery_store_cafe`, and `hotel_gym_spa`. Manipulable objects include apples, soda cans, bananas, bars of soap, tennis balls, bell peppers, mugs, bowls, cabinet doors, and refrigerator contents.

Asset preparation is fully automated. An offline baker decrypts the distributed USD when necessary, re-roots PBR texture paths, removes dataset-specific physics scenes and helper rigid bodies, and extracts articulated-joint metadata. It emits a plain USD and a compact specification (category/model identifiers, bounding box, base offset, canonical orientation, mass or density, openable links, semantic tags, and joints). At load time, the scene importer references this USD, applies the scene-JSON translation, scale, and quaternion (converted from $xyzw$ to $wxyz$), aligns the asset base with the

floor, registers its semantic roles, and installs its collision geometry. Adding a BEHAVIOR asset thus requires only baking and metadata validation, with no simulator-code change.

B.2 Implementation Details of Human Action Layer

B.2.1 Collision-Aware Autonomous Navigation

Stance selection. Navigation targets an *interaction-ready stance* rather than the object's centroid. Target grounding estimates a task-relevant contact point and samples floor-supported standing poses around it, each oriented toward that point and evaluated for (i) connectivity to the current pose, (ii) camera and full-body clearance, (iii) arm reachability, and (iv) collisions in both the idle and the anticipated arrival pose. The closest reachable valid stance is preferred. A blocked nominal stance can be snapped to a valid point within $0.60\,\mathrm{m}$; the last 4–6 frames then execute a short ($\approx 0.10\,\mathrm{m}$) approach to establish the final interaction distance.

Occupancy grid. Routes are planned on a $0.10\,\mathrm{m}$ ground grid rasterized from the same collision geometry used by manipulation, with separate layers for walls, furniture, door leaves, and floor support. Obstacles are included only where they intersect the human's traversed height band, i.e., 0.20–$1.90\,\mathrm{m}$ above the local floor for walls and 0.05–$1.90\,\mathrm{m}$ for furniture; cells without floor support are blocked. A Euclidean distance transform supplies the clearance $D(n)$ of each free cell. To avoid wrongly declaring narrow rooms disconnected, the grid is inflated by the largest radius in $\{0.20, 0.15, 0.10\}\,\mathrm{m}$ that still preserves a connected, reachable solution.

Search. We run eight-connected A* with diagonal corner cutting disabled and transition cost

$$c(n, n') = r\left[\ell_8(n, n') + \lambda \max\big(0, d_{\mathrm{safe}} - D(n')\big) + c_{\mathrm{door}}(n')\right], \tag{2}$$

where $r = 0.10\,\mathrm{m}$ is the grid resolution, $\ell_8 \in \{1, \sqrt{2}\}$ the step length, $\lambda = 8$, and $d_{\mathrm{safe}} = 0.35\,\mathrm{m}$. The clearance term biases paths toward open space without turning narrow passages into hard obstacles. For a closed door, the planner identifies the blocking leaf, routes to a collision-validated handle stance, executes the door-opening action, updates occupancy, and replans; if no such sequence exists, the action returns `PATH_BLOCKED` rather than teleporting the human. String-pulling simplifies the polyline only when a shortcut is collision-free and does not increase the weighted cost.

Execution and 3-D validation. The route is converted to an ACCAD gait rendered at 12 fps with a nominal speed of $1.1\,\mathrm{m\,s^{-1}}$. A heading controller uses a $1.2\,\mathrm{m}$ look-ahead, limits yaw change to 10° per frame, and blends toward the final object-facing orientation over the last $1.5\,\mathrm{m}$. Translation may pause during large turns, keeping the offset between body heading and travel direction below $\approx 35^\circ$. Finally, the articulated body, arm, and egocentric eye volumes are checked along the generated 3-D trajectory. If a cell that appeared free in 2-D fails this check, a temporary $0.20\,\mathrm{m}$ avoidance disk is inserted and the route is replanned; a bounded repair budget prevents infinite retries and yields a structured failure when no safe route exists.

B.2.2 Target-Driven Adaptive Manipulation

Manipulation is synthesized from a small set of reusable pose prototypes, target-relative end-effector curves, and online inverse kinematics (IK), rather than from a stored motion clip per object instance.

Contact and grasp selection. The target's oriented bounding box determines the contact region: broad, flat objects are approached near a reachable perimeter edge, and volumetric objects at a body- or face-centered target. Geometry also selects the grasp family: PINCH for small or flat objects, CYLINDRICAL when height exceeds $\approx 1.15\times$ the cross-sectional diameter, and ROUND otherwise. This estimate is considered jointly with stance selection, so navigation never commits to a stance from which the chosen contact is unreachable.

Body posture and arm IK. The body controller searches an adaptive posture ladder: counter-height targets keep an upright torso or mild lean; low targets use a $\sim 40^\circ$–86° forward lean coupled with hip, knee, and ankle flexion; and shelf insertion favors leg flexion with a more upright torso. A cheap arm-length bound prunes infeasible candidates. For the remainder, closed-form cyclic

coordinate descent adjusts collar, shoulder, and elbow rotations for up to 28 iterations, with a nominal $1\,\mathrm{mm}$ end-effector convergence target and action-specific final tolerances. If no body/arm solution exists, the action returns `OUT_OF_REACH`; a geometrically reachable but colliding approach returns `TRAJECTORY_COLLISION`.

Reach and tool trajectories. Every reach is expanded into pre-contact, contact, and retreat phases. The pre-contact offset is typically 0.10–$0.18\,\mathrm{m}$, shortened or redirected according to local geometry, and retreat height increases inside enclosures. Shoulder, elbow, wrist, forearm-midpoint, and metacarpal samples are checked throughout the curve. Action-specific trajectories are defined in the target frame: a repeated vertical stroke for cutting, a horizontal circle for stirring, a raster sweep for wiping, and rotation of the held source along a target-rim-centered path for pouring. IK and collision checks run at every sampled point before frames are committed.

Hand adaptation. The selected 45-D hand prototype is adapted to the live mesh by curling each finger in 6° increments using batched SMPL+H forward kinematics. A finger contact is valid within a $6\,\mathrm{mm}$ surface band with at most $4\,\mathrm{mm}$ penetration, and a grasp is accepted when at least four of five fingers make valid contact. Otherwise, the controller tries wrist perturbations within a natural range (yaw $\pm 10^\circ$, $\pm 20^\circ$; roll 0°, $\pm 15^\circ$), scores surface contacts, and resumes finger bending from the best candidate. Surface distance is queried from a padded voxel signed-distance cache in object-local coordinates, which makes the first encounter with a novel mesh costlier than subsequent grasps but avoids manually authored collision points. On successful contact, the simulator stores the object transform in the wrist frame,

$$ {}^{W}T_{O} = \left({}^{\text{world}}T_{W}\right)^{-1} {}^{\text{world}}T_{O}, \tag{3} $$

after which the object follows the wrist rigidly, without an artificial upright snap, until release.

Pose assets and extensibility. Pose data are deliberately compact. `hand/atlas/` contains three grip families, `side_wrap_family.npz`, `round_wrap_family.npz`, and `pinch_family.npz`, each storing a 45-D right-hand pose and optionally a family identifier, grip type, preferred world wrist rotation, and seat template. Optional object calibrations in `hand/atlas/objects/` currently cover cup, ball, box, plate, pen, and tool templates; they store build parameters and, when calibrated, an exact object-in-wrist transform. Objects without an entry derive their seat from the family template and measured cross-section, after which live finger wrapping adapts the pose to the instance. Supporting a new object therefore normally requires only baking its geometry and metadata, plus, if automatic seating is inadequate, a small calibration file with the desired wrist-relative transform. A genuinely new grip requires one family NPZ (e.g., from the prototype-generation utilities), registration of its identifier in `_PROTO_FAMILY`, and optionally a validated `grasp_hint` specifying the target region (`auto`, `center`, `edge`, or `body`), family (`auto`, `pinch`, or `wrap`), edge preference, height fraction, and clearance; no new full-body animation is needed. A new manipulation primitive provides its target-relative curve and declares whether each affected object is uncoupled, rides the end effector, is joint-driven, or remains body-anchored.

B.3 Implementation Details of Semantic Layer

B.3.1 Unified Action Interface

The public interface uses a deterministic grammar,

```
[Verb] <entity_name> (instance_id) [<entity_name> (instance_id) ...].
```
(4)

Commands are separated by newlines or comma–space delimiters, with bracket-aware splitting. Entity names are normalized to lower-case underscore form and resolved first by exact key and then case-insensitively; an optional resolver accepts uniquely grounded descriptions but rejects ambiguous references. The registry validates entity arity before compiling each command into one or more typed goals. Manipulations automatically insert a `NAVIGATE_TO` goal when needed, select a fresh nearest-reachable stance at execution time, and deduplicate redundant navigation. For example, a pick-and-place request is expressed as

```
[Walk] <red_mug> (2)
[Grab] <red_mug> (2)
[PutIn] <red_mug> (2) <drawer> (1)
```

The same request may also be given in the structured TaskPack form (fields `verb`, `args`, and optional `indices`, `scalar_arg`, and `place_spot`), which is rendered into the same textual grammar before execution.

Arguments and defaults. Entity arguments default to instance index 1, except symbolic connector arguments of `attach`, which default to index 0. The suffix `@place_spot=u:v` selects a normalized target-local location, with $u \in [-1, 1]$ from left to right in the target-facing frame and $v \in [0, 1]$ from front to back. `[Forward]` defaults to $0.6\,\mathrm{m}$; a unitless number denotes paces, an `m` suffix denotes meters, and displacement is capped at $20\,\mathrm{m}$. `[TurnLeft]` and `[TurnRight]` default to 90° and accept angles in $(0, 360]^\circ$, whereas `[TurnAround]` is fixed at 180°. `[Wait]` defaults to 3 s and accepts durations from 0.1 to 600 s.

Table 5 lists the 36 canonical primitives in the authoritative runtime registry, all marked implemented in the audited revision. Surface aliases are case-insensitive after normalization and map to the same primitive. A task-facing prompt may expose only a relevant subset without changing the executor's registered capability set.

Table 5: The 36 canonical action primitives, grouped by function. The second column lists the accepted surface verbs (written as `[Verb]` in a command, e.g., `[Walk]`) followed, in italics, by the arguments in order. Distance, angle, duration, and angle error are numeric; all other arguments refer to scene entities. "Held" means that the named object must match the persistent held-object state.

Primitive	Surface verbs and arguments	Operational semantics
Locomotion and posture		
`navigate_to`	`Walk`, `Run`, `WalkTowards` *target*	Plans and executes a collision-aware route to a live interaction stance near the target; `Run` is an alias and selects no separate speed controller.
`forward`	`Forward`, `WalkForward` optional *distance*	Collision-checks and executes a straight body-frame displacement.
`turn`	`TurnLeft`, `TurnRight`, `TurnAround` optional *angle*	Rotates in place while preserving any wrist-carried object; the optional angle applies to `TurnLeft` and `TurnRight` only.
`sit`	`Sit` *seat*	Lowers and anchors the body on the live seat top; records SITTING.
`standup`	`StandUp`	Reverses the seated pose, steps away from the seat, and clears SITTING.
`squat`	`Squat`, `SquatDown`	Lowers the body and egocentric camera into a crouch, then returns upright.
Grasping and placement		
`pick`	`Grab` *object*	Reaches, closes the adaptive hand, and establishes persistent held state; fails if another object is held.
`release`	`Drop`, `Release`	Releases the held object at its carried pose, makes it dynamic, and settles it.
`flip`	`Flip`, `TurnOver` *object*	Rotates the named held object by 180° without releasing it.
`place`	`Put`, `PutBack`, `PutObjBack`, `PutOn` *object, surface*	Places the named held object on a support, settles physics, and verifies ON.
`place_inside`	`PutIn`, `PutInside`, `Insert` *object, container*, optional *region*	Places the held object in a named/indexed interior region or an automatically distributed free region; verifies INSIDE.
`place_on_floor`	`PutOnFloor` *object, anchor*	Crouches, places the named held object on the floor beside the anchor, and verifies proximity.

Continued on next page

Table 5: (continued)

Primitive	Surface verbs and arguments	Operational semantics
`fall_to_floor`	`FallToFloor` *object, support anchor*	Applies a deterministic unattended transition from the stated support to the adjacent floor; rejects held, contained, or incorrectly supported objects.
Assembly		
`attach`	`Attach`, `Assemble`, `Install` *child, child connector, parent, parent connector*, optional *angle error*	Mates declared connectors at their 6-DoF target pose; the optional integer angle error perturbs the mating orientation for controlled evaluation.
`detach`	`Detach`, `Disassemble`, `Remove` *child, parent*	Removes the child–parent attachment and restores independent manipulation.
`raise`	`Raise`, `Erect`, `StandUpright` *assembly*	Rotates a flat assembly upright about its base, preserving internal attachments.
Fixtures and articulated objects		
`push`	`Push`, `PushRack` *fixture, destination*	With the hand empty, pushes a registered movable fixture to the destination; supported contents move with it.
`pull`	`Pull`, `PullRack` *fixture, destination*	With the hand empty, pulls a registered movable fixture to the destination, propagating contained objects.
`open`	`Open` *openable object*	Drives a registered swing or sliding joint to its open limit; verifies OPEN.
`close`	`Close` *openable object*	Drives the articulated joint to its closed limit; verifies CLOSED.
`open_door`	`OpenDoor` *door*	Reaches the handle, swings the door, and removes the opened leaf from navigation occupancy.
`close_door`	`CloseDoor` *door*	Closes the hinged door and restores its blocking occupancy in the navigation grid.
Devices and tool use		
`turn_on`	`TurnOn`, `SwitchOn` *device*	Physically presses the registered control and establishes RUNNING.
`turn_off`	`TurnOff`, `SwitchOff` *device*	Presses the control and clears RUNNING.
`cook`	`Cook` *food*	Requires the food to be in/on a suitable powered appliance, performs the interaction, and verifies COOKED.
`cut`	`Cut` *target, held tool*	Executes a collision-checked chopping trajectory; verifies CUT.
`stir`	`Stir` *target, held tool*	Executes a circular tool trajectory inside the target; the motion itself is rendered and logged.
`pour`	`Pour` *target, held source*	Tilts the held source around the target rim; for configured contents, verifies transfer into the target.
`wipe`	`Wipe` *tool, target*	Executes a raster sweep, acquiring the tool if the hand is empty; no separate cleanliness predicate is scored. The public order is tool first; the legacy target–tool order is recognized heuristically.
`use_computer`	`UseComputer`, `Type` *keyboard or workstation*	Optionally auto-sits, types with both hands, activates the display, and auto-stands if this action introduced the sitting.
`answer_phone`	`AnswerPhone` *phone*	Requires the named mobile phone to be held and moves it to the ear without navigating back to its original location; subsequent waiting preserves this pose.

Continued on next page

Table 5: (continued)

Primitive	Surface verbs and arguments	Operational semantics
Perception and control		
`look_around`	`LookAround`	Performs a level, full-circle egocentric scan.
`look_scan`	`LookScan`	Performs a floor-pitched sweep for nearby or low objects.
`look_at`	`LookAt` *entity*	Aims gaze at the entity; may reposition or reorient a label-bearing object to obtain a readable inspection view.
`wait`	`Wait` optional *duration*	Holds the current pose and renders stationary frames for the requested duration.
`confirm`	`Confirm`	Marks a human-policy halt/finalization point; intentionally emits no physical animation frames.

B.3.2 VERIFIABLE MULTI-TURN INTERACTION

The evaluation involves three participants: an assistant policy, a human-executor policy, and a persistent Isaac environment. Their information flow at turn t is

$$(I_{\le t}, u^H_{t-1}, \tau_t) \xrightarrow{\text{assistant}} u^A_t \xrightarrow{\text{human}} (c_t, u^H_t) \xrightarrow{\text{environment}} (I_{t+1}, r_t, e_t), \tag{5}$$

where I denotes sampled egocentric frames, τ the task instruction, u^A and u^H the two agents' utterances, c_t a grounded command, r_t the observed relation delta, and e_t a structured execution outcome. The environment persists across turns: object transforms, articulated joints, attachments, containment, the human pose, and held-object state are never reconstructed from language.

Reset and information boundaries. At reset, the evaluator loads a frozen task/runtime, resets or reuses a warm Isaac server, verifies that setup has not already satisfied any scored goal, executes an authored visual prologue, and pins its evidence-bearing opening frames. On each subsequent tick, the assistant receives only the sampled first-person visual segment, the task instruction, recent human speech, and a latest-segment marker; internal instance identifiers, relation tables, coordinates, held state, answer keys, and score are hidden. It responds with `$silent$`, an intervention prefixed by `$interrupt$`, an error report prefixed by `$error$`, or, in evaluation, the strict completion token `$done$` (equivalently `<done/>`).

The human executor receives the same visual evidence plus the currently visible/actionable object handles, held state, the assistant's utterance, the previous engine outcome, and the command grammar, and returns separate `<reasoning>`, `<action>`, and `<speak>` fields. Private reasoning and commands are logged but never shown to the assistant: only the human's speech, which is checked for accidental internal-ID leakage, and the pixels produced by execution cross this boundary. An action block may queue several lines, but only one command executes per tick and the assistant is polled between commands. In the TaskPack interaction runtime, a new intervention can preempt pending commands, and a parse or execution error flushes the remainder so that stale actions are not applied after a changed instruction.

Command execution and verification. Each command passes through five stages: (1) parsing and grounding under the fixed registry, recording whether the command was strictly accepted or repaired; (2) compilation into typed goals, including implicit navigation; (3) execution by the physical controllers described above; (4) physics settling and read-back of the actual relations and articulated state; and (5) postcondition validation and rendering of the next first-person segment. A failed primitive returns a structured `PrimitiveError` and no success frames. Even when a controller reports success, state-changing actions (open, close, power, cut, cook) are re-read from the live physical relations; a mismatch is rewritten as `POSTCONDITION_FAILED` and fed back to the human in plain language for recovery.

Scoring. The goal checker supports the physical predicates ON, INSIDE, HELD, OPEN, CLOSED, OFF, NEAR, SITTING, CUT, COOKED, RUNNING, ATTACHED, RAISED, and UNMOVED. Scoring reads the current physical relation set, positive goals, prohibitions, and successful trace outcomes.

Predicates already true before handoff are baseline-excluded, so agents receive no credit for the initial state. An assistant completion declaration stops the interaction but cannot manufacture success: the world is scored exactly as it stands.

Authored human policies. Authored multi-turn tasks may additionally use a fail-closed, acyclic `human-policy/1.0` state machine whose branches depend on timeout, world state, or the conjunction of assistant semantics and world state. A frozen semantic verifier maps assistant speech to an exact JSON tuple of intent, referents, and facts; the policy advances only if this tuple matches the authored transition and an independently queried world predicate agrees. Completion specifications can require alternative or negated physical states, exact relation counts, successful command multisets or sequences, timely interventions, and forbidden-before constraints, so persuasive but ungrounded dialogue cannot satisfy a task.

Logging. The evaluator keeps an append-only, crash-safe record of assistant and human prompts, raw and parsed outputs, silence, grounding/repair provenance, every attempted primitive and its arguments, actual relation deltas, required facts, success or error codes, frame intervals, wall time, score, and stop reason. These records make both physical execution and multi-turn communication auditable after an episode; a missing measurement is treated as unavailable rather than silently converted to zero.

The same loop supports the ASK, FIND, GUIDE, WATCH, NARRATE, and ORCHESTRATE interaction families. They differ in which participant initially holds information and when the assistant should intervene, while sharing the same physical executor, command registry, trace format, and postcondition-based verifier.

C IMPLEMENTATION DETAILS OF INTERACTIVEBENCH

InteractiveBench evaluates an egocentric assistant in closed loop rather than as a static video-question-answering model. In each episode, the evaluated Assistant receives chronologically ordered first-person-view observations and natural-language speech; and a separate Human executor converts the Assistant's utterances into environment actions. The benchmark comprises 120 frozen episodes in six equally weighted reported families of 20 tasks each. The unit of evaluation is a versioned TaskPack, which fixes the scene, entities, opening event history, public utterances, executable action programs, score contract, and provenance hashes.

C.1 BENCHMARK CONSTRUCTION

Construction. We construct each InteractiveBench task from a task family and a target capability. We first propose a causal scenario with a concrete goal or error to avoid, a role for the assistant, and actions the Human can perform. Two independent reviews check whether the proposal fits the family, requires assistance, can be enacted in a physical scene, and has a measurable outcome without help. Proposals that fail these checks are revised within a fixed budget or discarded.

Before choosing a scene, we define an assessment contract. It records the question being tested, the evidence the assistant must use, when help is needed, the information gap, the success condition, and the expected outcome without help. This contract remains fixed throughout construction. We then bind the scenario to an available scene, objects, and actions, checking that the chosen assets exist and preserve the intended test. Next, we author the initial world state, the events before the Human's request, the request itself, an oracle action sequence, a no-help sequence, and state-based goal checks. These elements form an executable task specification.

We run the event history and both action sequences through deterministic checks. They test action syntax, object affordances, hand occupancy, geometry, goal states, and whether the task is already complete or its answer is exposed too early. For repairable errors, we revise the scene binding, world state, and task script together, then rerun the checks. If a valid repair would change the assessment contract, we discard the candidate. A final, non-editing review checks that the task still requires the intended evidence, timing, and assistance.

Accepted specifications are compiled into TaskPacks and replayed in Isaac Sim. We execute the oracle, score the resulting physical state, and inspect the rendered views for visual errors. We also

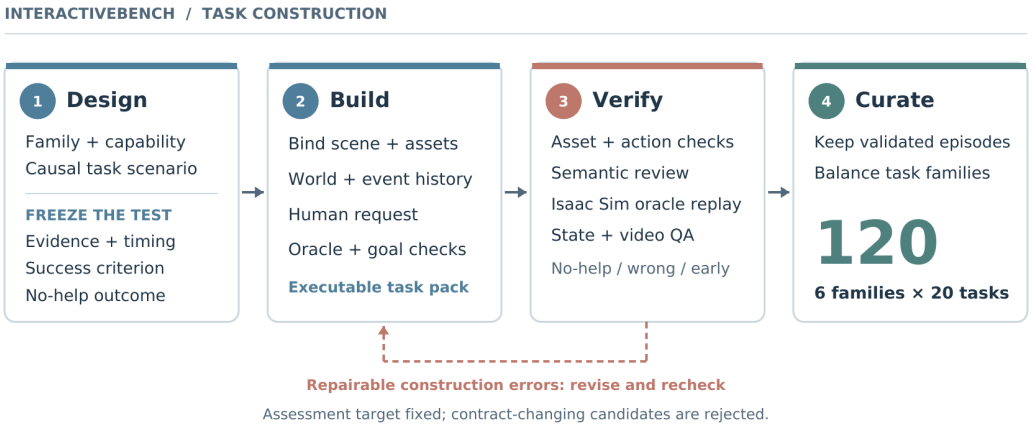


Figure 7: **InteractiveBench task generation.** For each task family and target capability, we define a scenario and fix its assessment target. We then build an executable task, check its semantics and actions, and validate it through Isaac Sim replay, visual inspection, and causal controls. The dashed arrow shows how repairable errors return for revision and revalidation without changing the assessment target. The final benchmark contains 120 tasks, with 20 per family.

test the interaction with scripted and model-based Humans, together with no-help, incorrect-help, and early-help controls. Failures caused by task content lead to a new task revision and another validation pass; simulator or infrastructure failures are fixed and retried without changing the task. Finally, we select 120 validated tasks, with 20 in each of the six families. We freeze the TaskPacks, code, prompts, scoring rules, and per-task revision records for reproducibility.

C.1.1 Design objective and candidate generation

Design objective. Construction targets interactive episodes rather than scripted renders. In a valid episode, (i) task-relevant evidence is genuinely visible from the wearer camera; (ii) this evidence can later leave the field of view or be occluded by intervening activity; (iii) the wearer asks for help, or proceeds toward a mistake, at a causally appropriate time; (iv) the Assistant can change the outcome through natural-language intervention; and (v) correct and incorrect assistance leave different verifiable world states. Requirement (v) excludes tasks whose success exists only in prose.

Candidate generation. Stories were generated per family. Up to six family workers ran in parallel, but generation within a family was serial, so each new proposal saw digests of previously accepted and rejected same-family proposals. This "avoid-from" context, containing both prior stories and rejection reasons, discouraged paraphrased duplicates and repeated design failures. The frozen authoring campaign used `gemini-3-1-pro-preview-tbd` as author and `gpt-5-5-genai-responses` as judge, with random seed 11, deterministic decoding (temperature 0), two independent review passes, and up to eight author revisions after rejection. The reviewer checked family fit, assistant necessity, physical expressibility, counterfactual discriminability, safety, and within-family causal diversity, and returned `KEEP`, `KEEP_DEBT`, or `RETRY`. `KEEP` admitted a story to compilation; it was not a benchmark pass.

C.1.2 Interaction assembly and release freeze

Interaction TaskPack. Each physical TaskPack is converted into a derived interaction TaskPack without changing its parent physical state. The derived layer adds a natural `prelude_say`, deterministic opening actions, a handoff utterance, frame selection, public and Human-only task contexts, spoken-cue contracts, WATCH intervention windows, and completion requirements. Opening actions are executed in the simulator and shown to the Assistant as pixels, never as a symbolic textual summary. The role-based information partition is described in §C.2.2.

Pre-sweep controls. Before entering a model sweep, each task's closed-loop assembly was challenged in a fixed order: (1) an oracle-verbal Assistant with a scripted Human checked wiring and

action anchors; (2) the same oracle with the model Human measured the executor ceiling; (3) silent, deliberately wrong, and deliberately early responders checked that the Human cannot solve the task independently and that failures are observable; and (4) a strong-model canary exercised free dialogue. The resulting audit separates eight layers—infrastructure, available evidence, Assistant decision, Human grounding, process/transition, world state, completion, and visual quality—so that a physically successful oracle video is evidence of feasibility, never a model score.

C.2 Evaluation Protocol

C.2.1 Two-agent closed loop

Each evaluation cell pairs one evaluated Assistant with a fixed Human executor and one simulator instance. In the documented configuration the Human model is `Gemini-3.1-pro`, held constant across compared Assistant models. An episode proceeds as follows.

1. **Load.** Load the exact TaskPack and verify its digest, family-specific interaction specification, scene assets, model route, frame policy, and fresh-process configuration.
2. **Opening.** Execute the deterministic opening. Each successful setup command yields a chronological FPV interval; setup failure invalidates the cell. Opening actions are unscored. The score baseline is captured after the opening, so a temporarily satisfied predicate cannot leak free credit.
3. **Assistant turn.** At interaction tick t, send the Assistant the newly selected FPV images, any new wearer speech, and the literal marker "Latest segment." The first turn also carries the common system prompt and family addendum. No private symbolic state is included.
4. **Parse.** Parse exactly one Assistant control response. Speech is passed to the Human. The response `$silent$` yields no speech, and `$done$` freezes the world for scoring; the freeze is deferred only while an already accepted Human action sequence is draining.
5. **Human turn.** Build a stateless Human turn from the public task context, current motor-grounding legend, held object, last action result, new Assistant speech, and any still-active instruction. WATCH additionally supplies the Human's private next no-advice action, which is never visible to the Assistant. The Human returns tagged reasoning, zero or more requested commands, and optional speech. The harness executes at most one command per simulator tick and polls the Assistant again after the resulting visual segment.
6. **Log.** Record the proposed and executed command, success/refusal receipt, frames and their provenance, structured visibility, held state, semantic state transitions, both agents' raw and parsed outputs, and latency. A rejected action is observable and must be recovered in a later turn; it is never silently converted into success.
7. **Terminate.** Continue until the Assistant declares done or the authored action/tick budget is exhausted. Goal satisfaction never auto-terminates the episode; otherwise a model could enumerate alternatives and be scored at the first lucky state. The final state and complete trace are then evaluated offline.

Frame selection. The FPV buffer preserves causal temporal structure. Each simulator command forms one interval, uniformly sampled to a configured number of frames while always retaining the true result frame; a `LookAt` interval additionally pins a `look_hold` frame so that the readable cue is not replaced by the release frame. The authored opening history is pinned or budgeted as specified by the TaskPack; later turns normally contribute only the latest interval. Stateless backends receive a bounded temporal selection that keeps the two most recent visual turns intact and samples older history uniformly, capped at 32 images in the documented configuration. All families supply chronological FPV evidence to the Assistant; NARRATE supplies no images to the Human.

C.2.2 Information boundaries and Human restrictions

Information is partitioned by role. The Assistant sees FPV images, wearer speech, its own dialogue history (or an explicitly declared memory treatment), and the public task request. It never sees simulator identifiers, the motor legend, the answer key, verification points, score, hidden completion state, oracle actions, or the Human's private WATCH plan. The Human sees only the information

needed to ground an Assistant instruction to an available action handle, and is explicitly forbidden to infer the answer, select a destination from the task semantics, or repair the Assistant's plan. The following family-specific restrictions are central to causal attribution.

- **ASK:** the Human cannot answer, count, read, calculate, or choose; only new Assistant speech may authorize one exact action. An unseen movable object requires an explicitly named search source; otherwise the Human asks for a visually grounded target and action.
- **FIND:** the Human cannot infer a remembered location or choose a search order. A movable target handle is exposed only while visible or held, so valid retrieval requires Assistant-directed search and visible-before-Grab grounding; this prevents semantic-ID auto-navigation from turning memory failure into physical success.
- **GUIDE:** the Human has no independent procedure. It executes the first currently actionable semantic step for one object, continues only its local motor consequences, and asks for the next step when idle.
- **WATCH:** without a motor instruction, the Human advances exactly one command along the private no-advice plan. A new explicit Assistant instruction preempts it; acknowledgements do not. Authored wearer speech is emitted at its fixed action cursor and never authorizes a motor act.
- **NARRATE:** the Human receives no images and cannot invent visual, audio, tactile, or other evidence. Private handles encode only an object or destination already identified by the Assistant, and at most one immediate action is returned.
- **ORCHESTRATE:** the Human knows the requested outcome but may not create, optimize, reorder, or repair the plan. It executes one strategic subtask and its local motor sequence, reports the result, and waits for the next subtask.

C.2.3 ASSISTANT PROMPTS

The box below reproduces the common Assistant system contract verbatim, apart from line wrapping. The user's task arrives through natural opening/handoff speech; ASK may additionally receive a public task-context paragraph. Neither the hidden answer nor a symbolic checklist is inserted.

Evaluated Assistant: common system prompt

```
You are an egocentric AI assistant helping a person through smart
glasses.
You receive the person's spoken request and first-person video in
chronological order.  After every environment action, you receive the
new visual segment and any sentence the person said.
You cannot see the evaluator's score, hidden completion state, answer
key, or simulator ground truth.  Judge progress only from the person's
speech and the first-person video you actually receive.
Choose exactly one response mode each turn:
-- $silent$ when speaking now would not help.
-- $interrupt$<one concise, natural utterance> for an answer, useful
next step, clarification, or timely instruction.
-- $error$<one concise, natural correction> when the person is
currently making a mistake.
-- $done$ only when you judge the requested task is complete.  The
world is scored immediately when you declare done, and nothing
afterwards counts.  $done$ must be the entire response:  never attach
an answer or instruction to it.  Speak first with $interrupt$, observe
the result, and declare done on a later turn.  Completion requires
positive evidence that the person's entire requested activity has
finished.  A quiet moment, harmless action, unchanged frame, or
absence of visible error is not completion; use $silent$ and keep
watching instead.
Use ordinary English.  Refer to objects by visible description and
location, never simulator ids.  Do not expose hidden reasoning.  Do
not claim that an unseen or undecidable fact is certain.
```

Assistant turn payload (schematic, in wire order)

```
[First turn only:  ``Before the setup actions, the person says:
<prelude>'']
<chronological FPV image blocks selected for this request>
[If speech occurred:  ``After the setup actions, the person says:
<handoff>'' or ``The person says:  <live speech>'']
Latest segment.
```

C.2.4 HUMAN EXECUTOR PROMPTS

The Human prompt is held fixed across Assistant models. The first box gives the common operative template; the production prompt adds family-specific anti-leak and malformed-action clauses with the same semantics. The second box shows the dynamically filled Human user turn.

Human executor: common prompt contract

```
You are the person wearing smart glasses and physically acting in the
world.  A separate Assistant watches your camera and talks to you.
You are cooperative and execute its instructions faithfully, but you
do not know the hidden answer key.  You cannot see the evaluator's
score, hidden completion state, or simulator ground truth; only the
Assistant may end the episode.
Ground every referenced object and destination before acting.  Act
only when the Assistant's words, available sensory evidence, and the
motor legend identify one command key.  If a reference is absent,
ambiguous, visually inconsistent, or cut off, leave action empty and
ask one short clarification.  Preserve the requested spatial relation
(inside → PutInside; on/onto → Put) and never substitute an object,
destination, or motor action.
A new Assistant instruction supersedes an unfinished old one.  Carry
one object at a time; consult the held-object state before Grab.  Walk
to a named destination before placement unless the latest receipt
says you are already there.  Read success/refusal receipts literally,
recover only as instructed, and never claim completion before every
requested clause succeeds.  Handle at most one primary object per
response.
Output exactly, in order:
<reasoning>two or three short private sentences</reasoning>
<action>zero or more grammar-exact simulator commands, one per
line</action>
<speak>one short spoken sentence, or empty</speak>
```

Human turn payload (schematic)

```
TASK CONTEXT: <public Human context>
TURN MODE: <opening / assistant-new / assistant-active / queued /
watch-default / idle>
[WATCH only:  private no-advice action plan and next cursor]
MOTOR REFERENCES: <currently grounded handles and public descriptions>
YOU ARE HOLDING: <object or nothing>
NEW ASSISTANT SPEECH: <utterance or none>
ACTIVE INSTRUCTION: <unfinished instruction or none>
[RESULT OF LAST ACTION: <success/refusal and reason>]
PRIVATE ACTION SYNTAX: <legal command grammar>
Reply with <reasoning>, then <action>, then <speak>.
```

The action language comprises navigation (Walk, Run, Forward, Turn, Look), manipulation (Grab, Drop, Put, PutInside, Push, Pull, Attach, Detach), open/close and power operations, and task actions such as Cut, Stir, Pour, Wipe, Cook, and AnswerPhone. Named arguments must use the exact current handle syntax. The harness extracts commands only from the `<action>` block, logs all proposed commands, and executes them one at a time.

C.2.5 VALIDITY AND CAUSAL ATTRIBUTION

Validity is audited before task scoring. A valid cell requires a successful scene/setup, the intended TaskPack hash and model route, a complete interaction trace, usable frames, and functioning Assistant, Human, transport, and simulator components. Infrastructure termination, corrupted or missing evidence, failed setup, or material Human answer leakage makes a cell VOID, to be rerun. Assistant silence, wrong guidance, loops, premature `$done$`, failure to declare, or incomplete work in an otherwise sound run is a valid model failure. Minor executor formatting repairs are logged; Human- or simulator-caused outcomes are never credited to the Assistant.

For each selected cell, the audit bundle retains the TaskPack and hash, `dialogue.json`, `eval.json`, the final manifest (or an equivalent terminal snapshot), all Assistant turns, `human_action_trace`, action receipts, structured visibility, opening and post-prologue baselines, final relations and persistent states, the held object, the stop reason, and the selected-attempt manifest. Attempt selection follows a fixed validity/provenance policy, never score.

C.3 EVALUATION METRICS

C.3.1 TASK-LEVEL METRICS

For a valid task t, the source-family evaluator returns strict completion $s_t \in \{0, 1\}$ and a task-normalized completion value $c_t \in [0, 1]$. For a reported family F of 20 tasks,

$$\mathrm{PassRate}(F) = \frac{1}{20} \sum_{t \in F} s_t, \qquad \mathrm{MCR}(F) = \frac{1}{20} \sum_{t \in F} c_t. \tag{6}$$

MCR denotes Mean Completion Rate; it is not an action ratio. Each task is normalized before receiving equal weight, so tasks with many objects, verification points, bundles, or capability atoms do not dominate.

C.3.2 CAPABILITY-LEVEL METRICS

Task outcomes and capability diagnoses are scored independently, using the frozen C1–C9 ontology in Table 6. Each task is annotated with its required capabilities, and each capability is assessed through separate evaluation points, referred to as capability atoms. A task may contain multiple evaluation points for the same capability.

Table 6: Capability ontology.

ID	Capability	Boundary
C1	Object/state recognition	Identify the current object and its current state; temporal updating belongs to C4.
C2	Spatial grounding	Locate objects, destinations, and relations in the egocentric scene.
C3	Scene-text reading	Transcribe and bind visible text; conclusions inferred from text belong to C6.
C4	Episodic memory/state tracking	Retain and update identities, events, locations, and states across time.
C5	Planning/replanning	Form, sequence, and revise a plan; Human execution cannot substitute for Assistant planning.
C6	Knowledge/reasoning	Apply rules, arithmetic, domain knowledge, and constraints to choose or infer.
C7	Error/risk detection	Recognize an error, omission, unsafe state, or impending risk.
C8	Intervention timing	Place the first task-relevant intervention between the frozen earliest-valid point and deadline; semantic correctness is scored elsewhere.
C9	Actionable communication	Produce unambiguous, parameter-complete, correctly ordered, physically executable language.

Evaluation-point scoring. Each required evaluation point receives exactly one label: `clean_pass`, `wrong`, `abstain`, `unobserved`, `assisted`, or `void`. Only `clean_pass` counts as a

passed evaluation point. Required points missed because of Assistant silence, premature stopping, or missed opportunities are labeled `unobserved`, not VOID. These points remain in the denominator and contribute no passes.

Equal-weight aggregation over evaluation points. Scores are computed separately for each evaluated model. For capability k, let E_k denote the set of task evaluations with complete required-point evidence. For each task $t \in E_k$, let $n_{t,k}$ be the number of required evaluation points for capability k, and let $p_{t,k}$ be the number labeled `clean_pass`. The capability score is

$$\mathrm{CapScore}(k) = 100 \times \frac{\sum_{t \in E_k} p_{t,k}}{\sum_{t \in E_k} n_{t,k}}. \tag{7}$$

Thus, we pool evaluation points across all included tasks and give each point equal weight. Tasks with more required points contribute proportionally more to the score; we do not average per-task pass rates. All capability scores are reported as percentages.

Independence from task completion. Capability points are scored using their own checks rather than the overall task outcome. Whole-task success is not propagated to all capabilities, nor is whole-task failure propagated as blanket capability failure. A failed task can therefore contain passed capability points, while task completion alone does not establish that every required capability was demonstrated.

C.3.3 Efficiency and reproducibility

Efficiency. Efficiency is computed only on strict-success episodes, against a task-matched shortest legal oracle. Each Human macro action (Walk, Look, Open, Grab, Put, etc.) counts as one step, including refused actions, which consume an interaction; speech, frames, the deterministic opening, and post-completion cleanup do not count. The stable-completion index is the earliest action after which all strict core conditions hold and remain true. Ratios are paired per task before aggregation; dividing a model's mean on one success set by an oracle mean on another is invalid. Efficiency is reported alongside, never folded into, MCR or Pass Rate.

Reproducibility. For each model–task cell, the released manifest preserves the fields `source_family` and `assigned_family`, the expected and observed TaskPack hashes, the hash origin, the rubric/version, the completion protocol, the evaluation status, and the evidence path. These fields ensure that a reclassification cannot silently change the scorer and that a run-level observed hash is not conflated with a frozen-rubric fallback hash. All aggregate tables are recomputable from the per-task and per-atom ledgers.

C.4 Task Catalog

Tables 7–12 list every released task, stating (i) what the wearer and Assistant must accomplish, (ii) the capability the episode isolates, and (iii) what the source-family evaluator verifies. "Baseline" denotes the state at the Assistant handoff, after the deterministic opening, not necessarily the state at simulator reset.

C.4.1 Guidance (20 tasks)

Guidance tests whether the Assistant can turn visible documents, domain knowledge, earlier observations, and numerical constraints into grounded, incremental instruction.

Table 7: Guidance tasks.

Task	Episode and capability tested	Evaluator validation
1	Read seven beverage labels and solve a globally coupled three-box allocation; tests OCR, arithmetic, and global constraint satisfaction.	Exact composites: A = milk-500 + water-100; B = milk-500 + juice-200; C = milk-500 + juice-200 + water-100. No wrong categories; all seven bottles form one exact partition.
2	Recall carton sizes seen before a phone interruption and fill three later-arriving orders; tests episodic visual memory and numerical allocation.	Exact size composites A = 500+100, B = 500+250, C = 500+250+100; all seven cartons partitioned once, no wrong sizes, and the phone preserved.

Task	Episode and capability tested	Evaluator validation
3	Archive obsolete briefing material before placing a new packet and pen; tests document-grounded ordered guidance.	Old material archived; new packet and pen on the reset table; new-item credit depends on removal of the obsolete material.
4	Use a stocking rule and price cards to arrange bread, candy, chocolate, and their labels in ascending-price order.	Three exact two-object composites occupy the low/mid/high promotional slots; none of the six objects remains on the cart.
5	Follow a tomato recipe through cutting, cooking, plating, and serving.	Tomato cut and safely cooked; cooktop off; the cooked form is the sole plated food; completed plate held; knife and spatula returned.
6	Sort four pantry goods by storage requirements; tests rule reading and storage knowledge.	Candy/chocolate exactly in the light display; coriander/coffee exactly in the closed dark cabinet.
7	Choose safe cooling methods for hot lemonade, purée, and an overheated phone.	Both foods refrigerated with fridge closed; phone on cooling counter with fan running; phone excluded from fridge.
8	Dry a wet towel and boot with material-appropriate equipment.	Towel alone in a closed, running dryer cycle; boot on or in the shoe rack and not in the dryer.
9	Apply a printed, tool-specific cleaning rule to six objects.	Mug and two bottles cleaned with blue sponge; camera/phone with yellow cloth; these five returned to the clean table; paper remains dry/unwiped and is filed.
10	Repair a three-object cyclic misplacement from a reset map.	Exact partition: phone in secure locker, water in fridge, folder on desk; ordinary door state is not independently scored.
11	Stock a capacity-limited shelf from three equal 800-mL cartons without exceeding capacity.	Exactly two cartons moved to the drink shelf and one remains at source; pre-existing milk/orange stock preserved.
12	Resume an interrupted breakfast plan after the ham stage.	Scone inserted and cooked, oven then closed; previously completed ham placement and burner-off state preserved.
13	Pack an IT return kit in dependency order.	Pad returned before headset; phone inside closed secure locker; exact box contents are pad/headset and exact locker content is phone.
14	Prepare a visitor table while respecting privacy/security ordering.	Staff cabinet closed; sign-in sheet, pen, badges on visitor table; spare folder in closed return locker; setup depends on privacy closure.
15	Resume an interrupted four-step afternoon-service plan after two completed steps.	Apple on setup table and clean mug on the vacated first supply table; completed water/towel states preserved; targets exact.
16	Resume a five-step hotel reset after the first two items.	Towel, soap, and toilet roll on bathroom shelf; already completed folder/water destinations preserved.
17	Close a tea station under storage and hygiene rules.	Milk refrigerated, tea dry-stored, paper discarded; station cleaned and cloth returned; mug placed only on the cleaned station with exact service contents.
18	Clear a cooking station and make it operational.	Used mug, paper bag, and bottle cleared to prescribed destinations; pan on cooktop and cooktop running, with activation gated by cleanup and pan placement.
19	Clear clutter, prepare and cook two vegetables, plate, and deliver a stir-fry.	Cleanup complete; both vegetables cut and cooked in the pan; cooktop off; exact finished contents plated at service table; preparation/cooking dependencies hold.
20	Select an exact one-litre, three-category tasting order from six labeled bottles.	Kit contains exactly GIN-200 mL, RUM-300 mL, and WINE-500 mL; the three larger decoys are excluded.

C.4.2 PLANNING (20 TASKS)

Planning tests strategic decomposition, resource and route optimization, constraint satisfaction, and replanning after interruption or state change.

Table 8: Planning tasks.

Task	Episode and capability tested	Evaluator validation
1	Pack an exact shipment from nine capacity-labeled packages; tests numerical planning and subset selection.	Exact water set {3 L, 3 L, 2 L}, juice set {4 L, 3 L, 2 L}, and 3-L lemonade; seven candidates exactly, excluding water-4 L and juice-5 L.
2	Execute a complete lunch order with schnitzel, broccolini, sealed sauce, tools, cooking, plating, and delivery.	Both foods cooked/browned and plated; cooktop off and spatula returned; exact two-food plate plus sealed sauce delivered to service pass.
3	Execute an ordered, multi-room housekeeping closeout.	Gift inside closed locker; water refrigerated; towel inside washer; washer closed and running with the towel.
4	Prepare a snack and replan when cutting reveals mold.	Moldy bread cut then discarded; fresh bread cut and plated with cheese/apple; safe plate at service pass; negative/exact constraints and knife return hold.
5	Balance three classroom desks with one water and one juice each using a short one-hand route.	Exact water counts across all three desks form one bundle and exact juice counts another; route length is reported separately.
6	Repair a three-way grocery misplacement with a continuous short route.	Rice to grains, barbecue sauce to sauces, and iced tea to beverages, one complete active-result bundle each.
7	Schedule rice and water before heat, then reset ingredients and tools.	Valid cooked porridge in pot on burner; rice bag in closed cabinet; knife in block; burner off; persistent recipe state records correct ingredient order.
8	Use a house map to plan folder-to-bed, phone-to-dining-table, and mug-to-kitchen moves with one necessary empty leg.	Three destination bundles complete; shortest-route quality is a separate efficiency measure.
9	Select, cut, plate, and deliver two allergy/low-sugar constrained fruit plates.	Six cut-fruit bundles on correct plates and two complete plate-delivery bundles; cheesecake, grape, and pineapple remain intact at baseline.

Task	Episode and capability tested	Evaluator validation
10	Minimize makespan for potato, pear, trout, and bread using one oven.	Four foods cooked on their own plates at the service table; pear cut; oven off and closed.
11	Parallel-schedule pork knuckle, trout, potato, and pear over oven and stove.	Four complete cooked own-plate dishes returned to prep table; pear cut; oven off/closed and stove off.
12	Detect an incorrectly stored cold brew, repair it, and complete a kitchen reset.	Cold brew and pineapple juice in fridge, mug in dishwasher, sponge in sink-side drying zone; fridge and cleaning cabinet closed.
13	Select the maximum feasible four of nine dishes under 30 minutes, one oven, and one stove.	Complete cooked plated baguette, pear, zucchini, and steak; pear cut; five unselected dishes unchanged; appliances safely reset.
14	Plan a short reset route for four objects across a five-room map.	Mug to kitchen, towel to bathroom, parcel to entry, and folder to living room, each as one endpoint bundle.
15	Filter nine dishes by four allergies, then maximize safe output under time/equipment constraints.	Complete cooked plated chicken, pear, zucchini, and steak; pear cut; all unsafe/unselected dishes unchanged; appliances safely reset.
16	Correct a scripted hotel-checkout mistake and finish a four-object reset.	Folder to desk, clipboard to wardrobe shelf, welcome box to entry console, and rinse mug to bathroom shelf; opening misplacement must actually occur.
17	Combine two delivery chains across a twelve-zone office map.	Folder to meeting room, agenda to copy room, package to reception, towel to bathroom, and phone to reception; one bundle per endpoint.
18	Infer seasoning order and coordinate chicken/bread roasting.	Persistent chicken state exactly soy-then-vinegar; complete chicken and bread plates at service counter; oven off and closed.
19	Recover a spoken plan after work and a phone interruption.	Phone returned from dining sideboard to living-room table; towel moved to kitchen; incident folder moved to the other living room; opening interference occurred.
20	Replan when a late spill appears during close-down.	Initial grime removed with sponge; apple on return cart; post-apple spill removed with towel and table dry; both tools returned.

C.4.3 ACCESSIBILITY (20 TASKS)

Accessibility models a blind or low-vision wearer. The Human executor receives no camera images, so the Assistant must provide timely, body-anchored, goal-relevant visual information. All tasks use CFSS-StateScore; "STG" in the table refers to the critical transition gates reported as STG-CFSS.

Table 9: Accessibility tasks.

Task	Episode and capability tested	Evaluator validation
1	Visually identify orange juice among a can, carton, and bottle and give body-anchored guidance.	Juice on breakfast table; soda and water remain at baseline (and never move under the transition gate).
2	Repeatedly reassess a growing gap between a pot and bowl, moving a scale only when it fits.	Scale on clearing counter; three bottles on holding counter; pot/bowl at baseline. STG keeps the scale at source until all bottles are cleared.
3	Compare remembered and current table settings and restore only a moved blue bowl.	Bowl restored to guest table; added coffee cup, unchanged wineglass/water, and all setting objects remain at handoff baseline.
4	Interrupt selection of a hazardous cleanser and redirect the wearer to baby oil.	Baby oil on side table; cleanser stably returned to cabinet, not held/on floor; bath salt at baseline. STG requires cleanser return before oil acquisition.
5	Admit that spray is absent from the current view, avoid a cactus, scan, and relocate the spray.	Spray on front cabinet; cactus, skin cream, and sponge at baseline; protected objects never move.
6	Prioritize putting down a carving knife safely before searching for a ringing phone.	Knife stably on prep counter and phone held. STG requires knife safety before leaving or acquiring the phone.
7	Read a NO GLASS sign, prevent wrong disposal, and locate the glass-return counter.	Bottle stably on return table, neither held nor in waste/on floor. STG forbids an unsafe intermediate disposal.
8	Filter nearby items by the semantic class "food," then compare distance to leaking deicer.	Apple in living-room waste bin; deicer, sponge, and salad at baseline.
9	Identify two food items among a non-food distractor and guide two refrigerator trips.	Apple and sandwich in closed fridge; sponge at baseline.
10	Detect an out-of-view mug/folder swap and distinguish it from wearer-caused changes.	Mug and folder restored to their respective baseline supports; phone and completed errand sheet preserved.
11	Track water-bottle counts across three visually similar tables.	Exact counts A=3, B=3, C=3, independent of bottle identity; STG prevents any table from exceeding three.
12	Detect broken-glass danger before placing a new blue cup.	New cup on hazard table; broken glass contained and broom returned; original cups unchanged. STG clears hazard before cup placement.
13	Select safe, genuinely open storage locations in a cluttered kitchen.	Apple in fridge or open counter; water in either; blue mug on open counter; fridge closed; six clutter objects preserved.
14	Read a cleanup guide and execute nested hazardous-waste disposal and ordinary cleanup.	Guide returned; glass inside yellow bag and bag inside red bin; apple/milk refrigerated; cup/bowl at sink; sponge unchanged. STG enforces bag-before-bin and hazard-first.
15	Remember a handwritten, three-item, cross-room note while ignoring distractors.	Apple in closed fridge, teapot on tea table, blue mug on side counter; white cup and sponge at baseline.
16	Search three laboratory regions for a phone, folder, and beaker.	Phone on instructor desk; folder and beaker each in either allowed native cabinet; coffee, sponge, marker, and water preserved.
17	Correct three sequential off-ticket picks while completing a catering ticket.	Apple, water, and sandwich on conference table; all eight off-ticket items restored. Intended STG returns each wrong pick before progress.

Task	Episode and capability tested	Evaluator validation
18	Inspect a refrigerator, detect a shoe hazard, and store milk safely.	Shoe on an allowed stable support outside fridge; milk inside; fridge closed. STG stages milk, removes shoe first, and closes last.
19	Recall where a phone was left after a long intervening kitchen activity.	Phone on the designated living-room coffee table; source-memory grounding is evaluated separately because endpoint state alone cannot prove recall.
20	Infer the remaining checklist items after two have already been completed.	Phone on office desk and badge on lobby desk; completed folder/mug endpoints preserved.

C.4.4 DECISION (20 TASKS)

Decision poses a concrete situated question whose answer has a physical consequence. It tests OCR, multimodal evidence integration, arithmetic, commonsense, state reconstruction, and the ability to turn an answer into an explicit action rather than relying on the Human to infer it.

Table 10: Decision tasks.

Task	Episode and capability tested	Evaluator validation
1	Choose the highest-calorie option without exceeding the corrected daily allowance; tests report reading, spoken correction, and constrained argmax $(2000 - 1780 - 110 = 110)$.	The 45-kcal chocolate syrup is on the breakfast table.
2	Fuse a children table and gifts-per-child table across rooms; compute $1 \times 2 + 2 \times 1.5 + 0 \times 1 = 5$.	Five gifts distributed exactly 2/2/1 among three cubbies; spare gift remains at supply.
3	Intersect an earlier remembered promo-brand set with twelve later router cartons.	AXO, BYT, and CUE cartons on display; all nine regular cartons at source/unmoved.
4	Reconstruct which items remain unfinished in a seven-point closeout after a long action chain.	Cabinet closed, oven off, mug/plate in cabinet, juice in fridge, can in trash, fridge closed.
5	Count stocked cartons, then consult a price sheet: $5 \times \$3 + 4 \times \$4 + 2 \times \$5 = \$41$.	\$41 invoice on approval desk; \$39 and \$45 alternatives remain at source/unmoved.
6	Aggregate two cabinet inventories, subtract removed items, and compute deficits against 4 towels, 5 mugs, and 7 waters.	Two replacement towels and one mug inserted; removed towel/water remain staged; both cabinets closed.
7	Bind an earlier Unit-B projector identity to a later A/B/C deployment table.	Unit-B projector on the specified pedestal; Unit-A/C distractors remain at source.
8	Combine two stock tables and replan after observing an occupied primary shelf.	Cat food reaches the DOG-OLD checkout counter; occupying dog-food can remains on its original shelf.
9	Recall which worktable item required immediate refrigeration and apply food-safety knowledge.	Raw trout in closed fridge; pasta and iced tea remain on worktable.
10	Identify the ingredient that makes dough rise from recipe context and functional knowledge.	Yeast on dough table; jam, bowl, flour, and water remain at sources.
11	Select a fruit for enzymatic tenderizing; distinguishes pineapple/bromelain from lemon acidity.	Pineapple on steak counter; lemon, steak, and bowl remain at source.
12	Recall a quantity of 40 and compare case sizes/prices: $5 \times 104 = 520 < 8 \times 70 = 560$.	Quote B on approval desk; Quote A remains at source.
13	Recognize the appropriate exercise apparatus and a safe spatial relation for a backpack.	Backpack released on floor beside parallel bars, not on the apparatus.
14	Choose the largest feasible snack under a remembered total-weight limit.	250-g cookie at lunch staging; coffee remains there; infeasible 351-g cracker stays at source.
15	Generalize an earlier category-level disposition rule to concrete stock items.	Milk and cheese in public trash; apple and cereal remain at source.
16	Add times across rooms and compare completion: option A ends 8:00, B ends 7:55.	Card B on bedroom cabinet; card A remains at source.
17	Subtract bookings from capacities across two sheets and choose the larger remainder (A=16, B=18).	Sheet B on bar; sheet A remains at source.
18	Use visual affordance and functional knowledge to choose a passive room fragrance.	Reed diffuser on entryway cabinet; atomizer and sponge remain at source.
19	Maintain an event ledger without looking back $(+1 + 1 - 1 + 1 = 2$ mugs), then complete a three-mug setup.	All three mugs on beverage station; sponge and phone at their required baseline locations.
20	Monitor a five-item grocery checklist across a phone interruption and detect that yogurt and feta were omitted.	All five listed items inside closed fridge; full-state check prevents repair from displacing already completed items 1/3/5.

C.4.5 PREVENTION (20 TASKS)

In Prevention, the wearer follows a private no-advice plan that is mostly correct but contains one or more consequential errors. The Assistant should remain quiet during correct work and intervene before each error's deadline. All tasks use WATCH-binary; intervention correctness and timing are measured separately by C7/C8.

Table 11: Prevention tasks.

Task	Error and capability tested	Evaluator validation
1	Cold cuts are moved before sanitizer is stored and the chemical cabinet closed; tests procedural order and food/chemical safety.	Mug in dish rack; towel in used sink; sanitizer in closed chemical cabinet; cold cuts on prep table and clean.
2	Adding another two-serving box would make seven, not exactly six; tests quantity tracking after a delayed update.	One- and three-serving boxes plus exactly one interchangeable two-serving box on service table; the other two-serving box at supply.
3	A used sponge remains on the table after the wearer reaches the break area; tests delayed omitted-step detection.	Sponge, mug, and towel in sink; phone on shelf; table physically clean.
4	A third wine and, later, chips violate "two wines, burger, no sides"; requires two interventions.	Exactly two interchangeable wines and burger on packing table; remaining wine and chips at stock.
5	Dough is about to enter the oven without an X score; tests prerequisite detection and recovery.	Dough cut and in oven; oven closed/running; current bake cycle began only after valid preparation.
6	A used sponge is confused with an identical clean sponge; tests event-bound identity.	Used sponge in used bin; clean sponge in closed clean cabinet; persistent used/clean attributes agree with history.
7	The fridge remains open while the wearer leaves to retrieve a phone; tests off-screen persistent-state monitoring.	Fridge closed; milk, raspberries, and phone all on breakfast table.
8	An oven door remains open when the wearer is about to start it; tests off-screen appliance state.	Pie in oven; door closed; oven running under a safe current cycle.
9	A towel previously used on a table is selected to dry a clean cup; tests instance provenance and hygiene.	Used towel at used station; clean towel returned to counter; cup clean and dry.
10	Before departure, a laptop and open window/fridge remain unresolved although the oven was already checked.	Laptop on departure console; window, fridge, and oven closed.
11	Among five similar towels, one used earlier is selected for a clean plate; tests long-horizon instance partition.	All five towels in their designated clean/used partitions; plate clean and dry.
12	A kitchen faucet remains on while the wearer leaves to answer a phone; tests persistent hazard tracking.	Sink off; phone on call table; rinsed apple on apple table; orange in fridge; plate in dishwasher; fridge closed.
13	Dough is loaded before the required drip tray; tests procedural prerequisite and recovery.	Tray in oven; dough properly supported by tray; oven closed; dough cooked under a valid cycle.
14	Mug and water are placed but a gift is omitted before checklist confirmation.	Mug, water, and gift all on the specified shelf.
15	An extra third scone and later an unlisted donut are added; requires repeated exact-order monitoring.	Exactly two interchangeable scones plus croissant, muffin, and brownie in box; remaining scone and donut at stock.
16	A folded towel is omitted from a guest-room checklist.	Water/folder on desk, towel in closed cabinet, mug on bedside table.
17	The wearer begins wiping before moving a phone and room key; tests order-sensitive cleaning.	Mug/towel at destinations; phone/key on personal table; sponge on cleaning shelf; desk cleaned after obstructions were removed.
18	A gift is placed before the desk is cleaned; tests order-sensitive housekeeping.	Mug, towel, and welcome card at destinations; desk clean; sponge on shelf; gift finally on desk.
19	An apple is taken to a prep table rather than the refrigerator; tests wrong-destination correction and checklist completion.	Folder at records counter; sanitizer in closed cleaning cabinet; apple in closed fridge; checklist at sign-off table.
20	A washer is about to start while a loose sock remains on a shelf; tests omitted-object and cycle integrity.	Sock and towel in closed/running washer; current cycle's frozen contents include both.

C.4.6 FINDING (20 TASKS)

Finding tests episodic object memory and evidence-grounded retrieval.

Table 12: Finding tasks.

Task	Episode and capability tested	Evaluator validation
1	Inspect labels to distinguish freezer from chiller, then classify five cold-delivery items.	Ice cream/lolly exactly in closed freezer; milk, yogurt, and lettuce exactly in closed chiller (GUIDE StateScore-v2).
2	Inspect contents of two otherwise ambiguous refrigerators to infer the dairy unit.	Cheese in secured dairy fridge; existing milk remains there, chicken in the other unit, and the other fridge remains closed (GUIDE StateScore-v2).
3	Track which of two identical glasses received vodka across a phone delay.	Alcohol-free glass on guest table; actual vodka-containing glass on adult counter, using persistent contents rather than hidden identity (WATCH binary).
4	Search three rooms for a sponge hidden in one of several similar low cabinets when its initial location was never shown.	Search exposes sponge before grab; sponge reaches result counter without hidden-ID navigation.
5	Retrieve the one of two identical erasers that touched a loose sheet rather than a folder.	Paper-contact eraser, not distractor, visibly acquired and placed in storage cabinet.
6	Recall the independent locations of two identical coconut-milk bottles.	Garden and fridge bottles directly retrieved to final tray; fridge opened before the hidden bottle is acquired.
7	Recall where sesame oil remained after another sauce was moved and a long distraction followed.	Sesame oil, not the other bottle, placed in target cabinet after visible acquisition.
8	Distinguish the gift left in the living room from an identical gift already visible in the kitchen.	Missing living-room gift, not visible distractor, retrieved to the kitchen result surface.
9	Recall mayonnaise on a counter versus similar jam and return it to the refrigerator.	Mayonnaise visible before grab and placed in opened fridge; jam not substituted.
10	Recall which of two similar bedside cabinets contains a thermometer.	Correct (left) cabinet opened; thermometer becomes visible and reaches infirmary desk; drink/right cabinet not substituted.

Task	Episode and capability tested	Evaluator validation
11	Search a bank of identical low cabinets for a lunch box whose initial cabinet is unknown.	Correct cabinet genuinely opened and lunch visible before grab; lunch placed on result worktop.
12	Track a mushroom inside a lidded portable box that was moved into a refrigerator.	Fridge opened, box retrieved, lid removed, mushroom independently extracted; mushroom and box both placed on dining table.
13	Recover the first and second of three identical milk cartons after different trajectories.	First and second cartons, not finished third carton, directly retrieved to the fixed work table.
14	Detect that the wearer searched a similar cabinet in the wrong room for a wallet.	Kitchen cabinet opened; wallet visibly retrieved to result counter; phone/wrong cabinet not substituted.
15	Retrieve the text-defined EQUIPMENT STORAGE form rather than the similar FOOD STORAGE form.	Equipment form visibly retrieved from opened file cabinet and placed with the food form on break-room table.
16	Predict and rediscover an apple that fell off-screen from a crowded table.	Floor apple visually rediscovered before grab and placed on result counter.
17	Recover a blue guest folder fully occluded by a larger brown cover.	Cover physically moved first; folder then visible, directly grabbed, and placed on nightstand.
18	Recall the red cola left on a dining table while blue water was refrigerated.	Red cola, not blue water, retrieved and put in opened fridge.
19	Retrieve a label-defined SPICED bottle from a lidded box while a WHITE bottle is visible.	Lid removed; SPICED bottle independently extracted and placed beside WHITE.
20	Retrieve black and blue squeegees nested in a box/cabinet while a white one is a distractor.	Cabinet and box opened, lid removed, both targets independently extracted; black and blue placed on their respective result tables.

D EVALUATION SETTING DETAILS AND MORE EXPERIMENT RESULTS

D.1 EXPERIMENT SETTING DETAIL OF ERROR TYPE ANALYSIS.

We developed the error taxonomy from the model results rather than defining it in advance. Each task family had its own success criteria. These criteria checked the final environment state and, when needed, the required action order or intervention time. For every valid failed episode, we reviewed the full conversation, visual frames, action results, and final state. We then found the earliest mistake that meaningfully caused the failure. Runs caused mainly by missing outputs, backend errors, or simulator problems were not treated as model errors.

We first wrote short descriptions of the repeated failure patterns and then merged similar patterns. This produced six main categories. R1 means that the model misreads a visible object or location. R2 means that it loses track of what happened or ignores an action result. R3 means that it applies the wrong rule or misses an important consequence. R4 means that it creates or updates the wrong multi-step plan. R5 means that it gives an instruction that cannot be executed in the current situation. R6 means that it speaks at the wrong time or fails to speak when help is needed.

Each failed episode received one primary error label so that it was counted only once in the distribution plots. Other problems could be recorded as secondary notes, but they did not change the primary count. Ambiguous cases were checked again using the original screenshots, model outputs, action feedback, and evaluator results. Therefore, the pie charts show the percentage of model-attributable failures assigned to each error type, rather than the percentage of all benchmark episodes.

D.2 EXPERIMENT SETTING DETAIL OF MEMORY AUGMENTATION.

We added a simple long-term memory to the assistant. At every step, the assistant produced both its next action and a new memory entry. The new entry was added to all earlier entries, rather than replacing them. The full memory was then included in the input for the next step.

The prompt asked the assistant to make each memory as detailed as possible. It was told to record useful facts such as object names, appearance, locations, states, spatial relations, visible text, and OCR results. It could also record earlier actions and their outcomes. This helped the assistant keep information that might be needed later.

We kept the visual input exactly the same as in the original setting. The assistant received at most 32 images at each step. When more than 32 frames are available, we retain the 8 most recent frames and uniformly sample 24 from the older history. Therefore, memory was the only change between the original and memory.

Evaluated Assistant: system prompt with self-managed memory

```
You are an egocentric AI assistant helping a person through smart
glasses.
You receive the person's spoken request and first-person video in
chronological order.  After every environment action, you receive the
new visual segment and any sentence the person said.
You cannot see the evaluator's score, hidden completion state, answer
key, or simulator ground truth.  Judge progress only from the person's
speech and the first-person video you actually receive.
Choose exactly one response mode each turn:
-- $silent$ when speaking now would not help.
-- $interrupt$<one concise, natural utterance> for an answer, useful
next step, clarification, or timely instruction.
-- $error$<one concise, natural correction> when the person is
currently making a mistake.
-- $done$ only when you judge the requested task is complete.  The
world is scored immediately when you declare done, and nothing
afterwards counts.  $done$ must be the entire response:  never attach
an answer or instruction to it.  Speak first with $interrupt$, observe
the result, and declare done on a later turn.  Completion requires
positive evidence that the person's entire requested activity has
finished.  A quiet moment, harmless action, unchanged frame, or
absence of visible error is not completion; use $silent$ and keep
watching instead.
Use ordinary English.  Refer to objects by visible description and
location, never simulator ids.  Do not expose hidden reasoning.  Do
not claim that an unseen or undecidable fact is certain.
SELF-MANAGED APPEND-ONLY MEMORY MODE. Maintain a chronological private
memory ledger across turns.  Each request contains at most 32 images
selected from the visual history.  The two latest visual turns are
kept intact, while older frames are uniformly sampled.  Some earlier
frames may therefore disappear.
The output-format instructions below replace only the earlier
formatting rules.  The meanings of silent, interrupt, error, and done
stay unchanged.  Return exactly these two XML elements in this order,
with no text outside them and no Markdown fence:
<response>one valid control response:  $silent$, $interrupt$...,
$error$..., or $done$</response>
<memory>only the complete new factual information learned on this turn,
or empty</memory>
Memory is private and is never spoken to the person.  The system
appends each non-empty update to the chronological memory ledger.
Never write your own ``[Memory entry N]'' label.  Do not repeat or
rewrite facts that are already in the memory.  If this turn adds no
useful new fact or correction, return an empty <memory></memory>.
Make each new memory entry as detailed as possible.  Record
task-relevant object categories, colors, materials, shapes, sizes,
counts, and distinctive marks.  Record every visible label or OCR
string exactly as shown, including capital letters, numbers, and units.
Use [ILLEGIBLE] or [OCCLUDED] instead of guessing.  Also record object
and person states, precise locations, events, completed actions,
observed results, unresolved questions, and required follow-up.
Use these sections only when they contain new information:
EVENT: actor | action | source -> destination | directly observed
result
CURRENT_STATE: object | state | current location | evidence
HELD: actor | object or empty | evidence
PENDING: unresolved required follow-through | last confirmed step |
next check
UNCERTAINTY: claim | alternatives | evidence needed
Use short lines and omit unused sections.  When a pending item is
```

```
resolved or an object state changes, add an explicit resolution or
correction.  The newest evidence wins.  Keep the location of an event
separate from the object's current location.
One action proves only its own result.  Walking proves a location
change, and grabbing proves possession.  Neither proves placement,
closing, turning something off, or task completion.  A stated plan
or destination also does not prove that the action happened.  Prefer
evidence in this order:  a successful action result or direct
post-action view; a current explicit statement from the person;
older visual evidence or memory; and inference.  Mark conflicts and
uncertainty instead of treating weak evidence as certain.
Before returning $done$, check every PENDING item against later
evidence.  Every required physical change must have a successful
action result or direct post-action evidence.  Confirm that no task
object remains held unless the request requires it.  Restore any door,
lid, or other object moved for access unless the request requires a
different final state.  If any required placement, closing, turn-off,
release, or restoration is not confirmed, keep it PENDING and do not
return $done$.
Do not put control tokens, hidden reasoning, simulator ids, answer
keys, or instructions to the person inside memory.  The complete
accumulated memory ledger is provided again on the next turn and is
never truncated by the system.
```

D.3 Additional Experiments about Comparative of Dynamic vs. Static

We made a static version of the same 120 tasks used in the interactive benchmark. The final benchmark has six task families: Guidance, Planning, Accessibility, Decision, Prevention, and Finding. Each family has 20 tasks. We used the same scenes and recorded first-person episodes as the interactive benchmark, so we did not create easier scenes for the static test. We stopped each episode just before the assistant had to answer or make a decision. For a Prevention task, we stopped after the warning cue appeared but before the assistant could intervene.

The static model saw a fixed sequence of images instead of a live environment. Each image sequence was paired with one short question. The question kept the main goal of the original dynamic task. A Finding item asked where an object was. A Guidance or Accessibility item asked for the next useful instruction. A Decision item asked for the answer or choice needed at that point. A Prevention item asked whether the user should be stopped and what was wrong. A Planning item asked for the full plan. We wrote the reference answer from the original task definition and the correct reference run, but only used facts available before the cut. The model did not see the reference answer, scoring rule, future frames, action results, or later user feedback. It also had no chance to act, check the result, or recover from an error.

We evaluated eight models once on every static item. A frozen semantic judge compared each response with the reference answer. Exact wording was not needed; the answer passed when it gave the same key fact, location, instruction, warning, or plan without a harmful contradiction.

Table 13: Static accuracy across the six task families. Every entry uses a fixed denominator of 20 tasks. Static items without enough evidence are counted as failures.

Model	Guidance	Planning	Accessibility	Decision	Prevention	Finding
GPT-5.5	45%	50%	50%	75%	70%	55%
GPT-5.6 Sol	45%	50%	60%	70%	75%	50%
GPT-5.6 Luna	35%	50%	55%	60%	65%	40%
Claude Opus 5	40%	45%	50%	70%	70%	50%
Claude Fable 5.1	45%	65%	65%	70%	75%	50%
Claude Opus 4.8	35%	30%	25%	60%	45%	40%
Gemini 3.1 Pro	60%	50%	55%	50%	65%	30%
Muse Spark 1.1	45%	45%	60%	65%	75%	60%

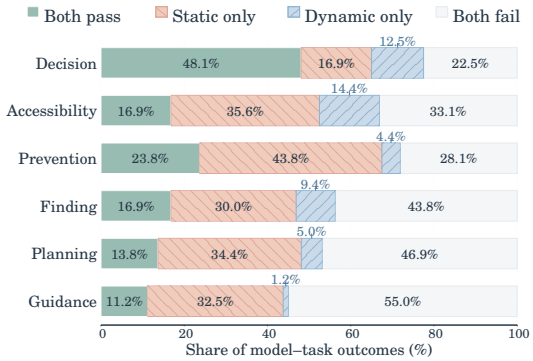


Figure 8: **Performance on Static Adaptations of InteractiveBench** inflate 8 models' scores ; large Static Only (red) and Dynamic Only (blue) regions show a severe evaluation gap.

Examples of the dynamic-to-static conversion. The following examples show how we kept the task goal but removed interaction.

- **Decision (ASK-13011).** The dynamic task asks the assistant to read a price sheet and approve a milk order. The static item shows all frames before the question and asks for the total. The reference answer is $41: five VAN cartons at $3, four CHOC cartons at $4, and two ORIG cartons at $5.
- **Finding (FIND-14001).** The dynamic task asks the assistant to find and bring two coconut-milk bottles. The static item asks where the two bottles are. The reference answer says that one is in the kitchen refrigerator and the other is on the outdoor table in the garden.
- **Guidance (GUIDE-90002).** The dynamic task asks the assistant to guide the user through preparing an area for visitors. The static item keeps the viewed checklist and asks what must happen first. The reference answer is to close the open green staff locker because the checklist puts privacy first.
- **Accessibility (NARRATE-87016).** The static input shows the cleanup guide, broken glass, and nearby tools. It asks for the next useful instruction. The reference answer says to handle the broken glass first and place the shard in the open yellow paper bag, without using bare hands.
- **Planning (ORCHESTRATE-14008).** The dynamic task asks the assistant to guide a multi-room move of a folder, a phone, and a mug. The static item asks for the complete efficient plan. The reference route is: move the folder to the bed, take the phone from that bed to the dining table, walk empty to the other living room, and move the mug to the kitchen counter.
- **Prevention (WATCH-16005).** The user says that the dough is ready and will go straight into the oven. The static item asks whether the assistant should interrupt. The reference answer is yes: score a shallow X in the dough first, return the knife, then put the dough in the oven and close the door.

D.4 Additional Experiment of Task Horizon and Model Performance

In this experiment, we asked a simple question: do models succeed less often when a task requires a longer sequence of actions? We used 120 tasks from six task families. We evaluated eight models on each task. We measured task length using a reference solution, rather than the number of actions taken by the model. This avoids a major bias: a failed model may repeat actions or run until the step limit, while a successful model may stop early. Let B_t be the best available reference action count for task t. We normalized this count separately within each task family:

$$h_t = \frac{B_t - \min_{j \in \mathcal{F}(t)} B_j}{\max_{j \in \mathcal{F}(t)} B_j - \min_{j \in \mathcal{F}(t)} B_j},$$

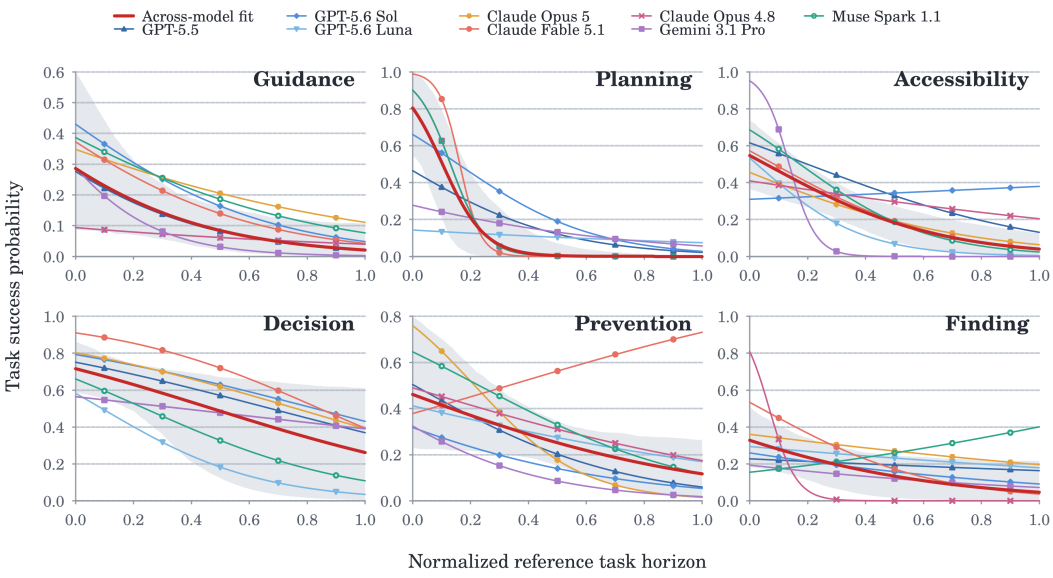


Figure 9: Complete-task success probability versus normalized reference task horizon across six task families. Each colored curve is a model-specific Firth logistic fit. The red curve is the pooled fit across models, and the gray band is its pointwise 95% task-bootstrap interval. Horizon is normalized separately within each task family. The figure compares different tasks and does not represent progress within a single episode. No monotonicity constraint was applied.

where $\mathcal{F}(t)$ is the family containing task t. Thus, $h_t = 0$ represents the shortest reference task in that family, and $h_t = 1$ represents the longest. These values are relative to each family. The horizontal axis compares different tasks; it does not show progress within one episode.

For visualization, we fitted a separate Firth logistic regression for every model and task family. The colored curves in Figure 9 show the fitted success probability of each model. The red curve is the pooled fit across models, and the gray band is its pointwise 95% task-bootstrap interval. The curves were not forced to decrease. For the main statistical result, we first averaged the eight binary model outcomes for each task and then fitted a task-level linear probability model. The overall model included task-family fixed effects. During bootstrap and permutation tests, all eight outcomes from the same task were kept together.

The family-adjusted slope was -0.431, corresponding to a 43.1-percentage-point decrease from the shortest to the longest normalized horizon (95% bootstrap CI: -58.1 to -28.9; one-sided permutation $p < 0.0001$). All six family-level point estimates were negative, and 45 of the 48 model–family curves also decreased. However, not every family-level test was individually significant. These results show an association between longer tasks and lower complete-task success, but they do not prove that task length directly causes failure. Longer tasks may also require more search, more object interactions, more state changes, or stricter safety constraints. Some labels were reconstructed from saved traces, so we treat this result as provisional.

D.5 Additional Human-Agent Validation with Real Human Participants

Purpose and setup. We examine whether replacing a model-based Human agent with a human participant changes the evaluation of an Assistant. We compare four model-based executors—Fable 5.1, Muse 1.1, GPT-5.6, and Gemini 3.1—with three human participants in InteractiveWorld. The comparison covers the same 36 tasks, with six tasks from each of the six benchmark families. Each executor is paired with each of four Assistants. The human participants take the executor role in the simulator; this is not a physical-world deployment study. We report the number of fully completed tasks out of 36 for each Assistant–executor pairing.

The model-based Human agents follow the instruction-following protocol specified in Appendix C.2.4. This prompt asks them to carry out the Assistant's guidance, ground object references before acting, and request clarification when an instruction is ambiguous. The comparison therefore concerns prompted agents in this defined executor role, rather than unrestricted agents pursuing their goals.

Table 14: **Assistant performance with model-based and human executors.** Rows identify the evaluated Assistant; columns identify the executor. Each entry is the number of successful tasks out of the same 36-task subset. Human 1–3 are three individual humans, not three independent task sets.

	Model-based Human agent				Human participant		
Assistant	Fable 5.1	Muse 1.1	GPT-5.6	Gemini 3.1	Human 1	Human 2	Human 3
Fable 5.1	13/36	15/36	13/36	13/36	15/36	16/36	15/36
Muse 1.1	9/36	9/36	8/36	9/36	10/36	11/36	9/36
GPT-5.6 Sol	8/36	9/36	11/36	10/36	8/36	11/36	9/36
Gemini 3.1	8/36	8/36	5/36	7/36	8/36	8/36	7/36

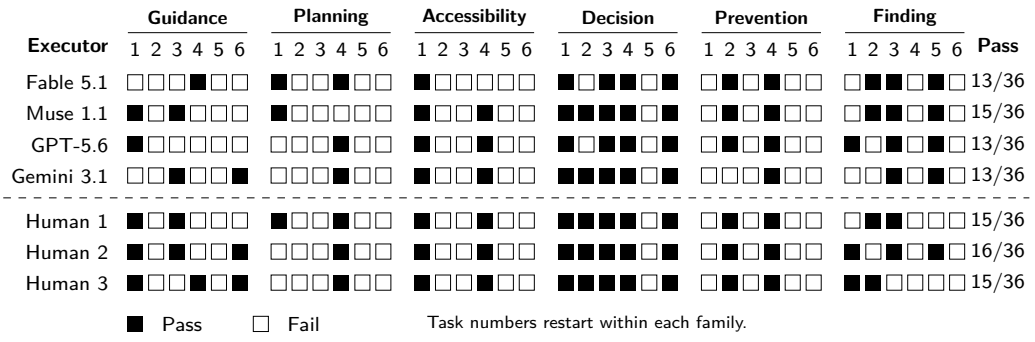


Figure 10: **Task-level outcomes for Fable 5.1 with different executors.** Rows are executors and columns are the 36 tasks, grouped into six families. Filled squares indicate success and open squares indicate failure. Task numbers restart within each family. The dashed line separates model-based Human agents from human participants.

Executor changes preserve the broad performance pattern. Table 14 shows that Fable 5.1 is the best-performing Assistant with every executor, while Gemini 3.1 is lowest or tied for lowest. Switching among the four model-based Human agents changes each Assistant's pass count by at most three tasks out of 36, or 8.3 percentage points. Across all seven executors, the largest observed range is three tasks, or 8.3 percentage points. Executor choice therefore affects individual scores, but does not overturn the broad performance pattern in this subset. The ordering of Muse 1.1 and GPT-5.6 Sol does change across executors, so we do not claim that the complete ranking is invariant.

Pass rates are close to those obtained with human participants. For Gemini 3.1, the Human agent used in our main evaluation, the four Assistants achieve pass rates of 36.1%, 25.0%, 27.8%, and 19.4%, in the row order of Table 14. Averaging over the three human participants gives 42.6%, 27.8%, 25.9%, and 21.3%, respectively. The human-minus-agent differences are therefore $+6.5$, $+2.8$, -1.9, and $+1.9$ percentage points. The observed scores are therefore relatively close, although the executor-related differences are not zero.

Agreement extends to individual task outcomes. To check whether similar overall pass rates hide different successful tasks, we inspect the 36 task-level outcomes for Fable 5.1 as the Assistant (Figure 10). The Gemini 3.1 executor agrees with Human 1, Human 2, and Human 3 on 30/36, 33/36, and 28/36 tasks, respectively. Their mean outcome agreement is 84.3%. For comparison, the three human–human pairs agree on 31/36, 30/36, and 31/36 tasks, giving a mean of 85.2%.

Thus, for this Assistant, agent–human agreement is close to the agreement observed between the human participants themselves. Agreement counts both shared successes and shared failures; it measures consistency of task outcomes, not identical actions or dialogue. These descriptive comparisons reuse the same tasks and are not independent samples or a statistical equivalence test.

Implications and scope. Together, these results provide preliminary support for prompted Human agents as a practical proxy for human instruction following in InteractiveWorld. Both the broad Assistant comparisons and the detailed Fable 5.1 outcomes show correspondence with evaluations involving real participants. This addresses whether the benchmark results depend entirely on a particular simulated executor.

E FUTURE WORK

During this study, we identified several interesting directions to explore next. (1) Since InteractiveWorld provides verifiable, state-based rewards, we will build a reinforcement learning platform on it to train interactive abilities efficiently. (2) We will make the Human agent behave more like real people, through training or other means, and simulate users with different personalities and skill levels. (3) As InteractiveWorld supports scanned real-world scenes, we will bring in more real environments and situations where people need an assistant, making the tasks more realistic.