

# AGENTHANDS: Generating Interactive Hand Gestures for Spatially Grounded Agent Conversations in XR

Ziyi Liu  
liu1362@purdue.edu  
Google XR Labs & Purdue  
San Jose, CA, USA

David Li  
davili@google.com  
Google XR Labs  
San Jose, CA, USA

Zhongyi Zhou  
zhongyizhou@google.com  
Google XR Labs  
Tokyo, Japan

David Kim  
kidavid@google.com  
Google XR Labs  
Zurich, Switzerland

Ruofei Du  
me@duruofei.com  
Google XR Labs  
San Francisco, CA, USA

Xun Qian  
xunqian@google.com  
Google XR Labs  
San Jose, CA, USA

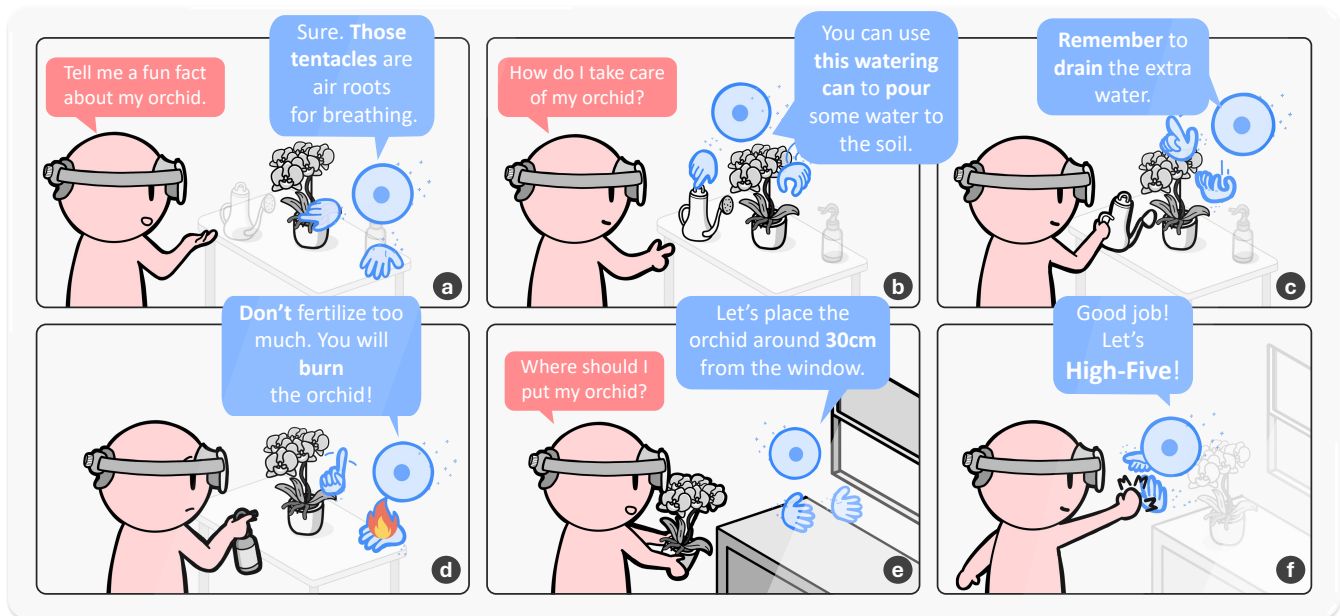


Figure 1: Overview of AGENTHANDS, an end-to-end system empowering an AI agent with expressive, interactive hands for spatially grounded conversations in XR. In this orchid-care scenario, an XR user asks about a new plant; (a) AGENTHANDS animates the agent to move towards the orchid and point directly to its aerial roots while sharing a fun fact. (b) Leveraging scene understanding, AGENTHANDS locates the watering can and performs a ‘pouring’ gesture as the explanation unfolds. (c) To reinforce the concept of drainage, a ‘beat’ gesture with the right hand draws attention, while the left hand illustrates holding and draining the pot with a cylindrical proxy. (d) For behavioral directives like “don’t over-fertilize”, the agent pairs a ‘negation’ gesture with a visual effect to convey risk. (e) When numeric guidance is needed (“30 cm from the window”), it performs an in-situ measurement gesture to make the distance concrete. Finally, (f) the agent celebrates successful completion with an interactive high-five with the user.

## Abstract

Communicating spatial tasks via text or speech creates “a mental mapping gap” that limits an agent’s expressiveness. Inspired by

co-speech gestures in face-to-face conversation, we propose AGENTHANDS, an LLM-powered XR system that equips agents with hands to render responses clearer and more engaging. Guided by a design taxonomy distilled from a formative study (N=10), we implement a novel pipeline to generate and render a hand agent that augments conversational responses with synchronized, space-aware, and interactive hand gestures: using a meta-instruction, AGENTHANDS generates verbal responses embedded with *GestureEvents* aligned



This work is licensed under a Creative Commons Attribution 4.0 International License. CHI '26, Barcelona, Spain

© 2026 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-2278-3/2026/04  
<https://doi.org/10.1145/3772318.3790938>

to specific words; each event specifies gesture type and parameters. At runtime, a parser converts events into time-stamped poses and motions, driving an animation system that renders expressive hands synchronized with speech. In a within-subjects study (N=12), AGENTHANDS increased engagement and made spatially grounded conversations easier to follow compared to a speech-only baseline.

## CCS Concepts

• **Human-centered computing** → **Mixed / augmented reality; Natural language interfaces.**

## Keywords

co-speech gestures, conversational agents, extended reality

### ACM Reference Format:

Ziyi Liu, David Li, Zhongyi Zhou, David Kim, Ruofei Du, and Xun Qian. 2026. AGENTHANDS: Generating Interactive Hand Gestures for Spatially Grounded Agent Conversations in XR. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain*. ACM, New York, NY, USA, 24 pages. <https://doi.org/10.1145/3772318.3790938>

## 1 Introduction

On mobile devices, general-purpose AI assistants began as command-driven tools for completing digital actions (e.g., Siri [29]) and controlling smart-home devices (e.g., Google Assistant [44], Alexa [2]). They have since evolved into multi-modal, large language models (LLMs) powered conversational agents such as ChatGPT [53] and Gemini [43]. These agents sustain dynamic dialogue and increasingly incorporate signals about the user’s physical context. For instance, in a grocery aisle, a user could hold the smartphone in front of two yogurts and ask, “Which of these two yogurts is better?”; in a museum, a user could ask “What’s important about this painting?”

However, despite these advances in contextual awareness, these agents primarily deliver text and speech based responses that struggle with spatial communication. Consider a user asking, “How do I use this 3D printer to print something?”. The agent might respond with lengthy textual instructions displayed on the mobile device such as “First, locate the power button on the back side of the machine, then navigate through the menu using the control panel...” Such spatially grounded conversations force users to mentally translate abstract verbal descriptions into concrete spatial operations, creating a cognitive burden that leads to confusion, errors, and ultimately undermines the intuitive human-AI interaction [11, 21].

Extended Reality (XR) offers a promising platform to address this spatial mapping limitation by bridging digital and physical worlds, overlaying virtual content directly onto physical spaces while sensing environmental context through built-in cameras and sensors. Recent research has explored LLM-powered XR assistants for various applications including augmented object intelligence [12], object information retrieval [38], and visually-enriched task guidance [74]. Yet these systems remain task-specific and predominantly rely on text responses or virtual UI overlays that require additional cognitive effort to interpret [1, 8]. Moreover, UI arrows and panels convey spatial references but not the social cues (acknowledge, caution, encourage) that sustain attention and trust in situated tasks. They

also rarely synchronize with evolving, LLM-generated responses, which diminishes the sense of engaging with a vivid, present agent. Consequently, the core challenge remains to design a versatile XR agent that can fluidly translate open-ended dialogue into interactive, emotionally attuned, and spatially grounded behaviors.

Human communication naturally integrates hand movements that enhance comprehension and engagement. These include pointing to direct attention and to establish shared spatial reference [33], hand movements that demonstrate actions or represent objects [19, 49], and social gesture cues such as thumbs-up [13] and v-sign [16]. These hand-based patterns show that physical movement can convey spatial information, demonstrate procedures, and sustain engagement—capabilities largely absent from today’s text-based AI assistants.

Drawing inspiration from hand-based communication, we present AGENTHANDS, an LLM-powered XR system that augments verbal agent responses with expressive and interactive co-speech hand gestures. Guided by a taxonomy of the agentic hand behaviors distilled from a formative study, we design a library of hand gesture behaviors and composition rules. Conditioned on a custom system prompt, AGENTHANDS augments agent responses by generating *GestureEvents* time-aligned to their trigger words, then uses them to drive a parametric animation engine that renders context-appropriate co-speech hand movements. Meanwhile, leveraging the spatial reasoning of LLMs, we propose a lightweight pipeline to achieve object-level scene understanding, unblocking robust spatial anchoring of the generated hand gestures. The resulting spatially grounded conversations with the agent transform passive text/speech consumption into active, embodied interaction, improving knowledge transfer, reducing ambiguity, and enhancing the user experience.

In summary, we highlight following contributions:

- (1) A set of **hand agent attributes** and a **multi-dimension taxonomy**, distilled from a formative study (N=10), that guides the design of co-speech hand embodiment for spatially grounded conversations in XR.
- (2) **AGENTHANDS**, an LLM-based, end-to-end system that augments verbal/textual responses with expressive, spatially anchored, and interactive hand gestures, improving user engagement and the clarity of agent responses.
- (3) **Empirical evaluation and research outlook** from a within-subjects comparative user study (N=12) that demonstrates the benefits of AGENTHANDS while surfacing insights for future research.

## 2 Related Work

### 2.1 General-Purpose Conversational Assistants

General-purpose AI assistants such as Siri [29], Google Assistant [44], and Alexa [2] have long been integrated into daily routines. Users interact with these agents in a command-driven manner, prompting them to perform pre-defined tasks such as checking weather, controlling smart home devices, or initiating navigation [28, 45].

Recent advancements in large language models (LLMs) have introduced a new era of AI assistants. Unlike traditional command-based systems, conversational agents such as ChatGPT [53] and Gemini [43] engage users through dynamic dialogue and ground

language in the physical world by recognizing objects, spatial relations, and task context. These versatile agents have become widely adopted across everyday contexts, from cooking guidance to fixing home electronics. Along this paradigm, users take the role of information receivers who interpret and use the assistant’s responses; however, the conversational medium means responses remain mostly text-based despite advances in context awareness.

Scientists have studied the shortcomings of text-based responses. When handling spatial tasks in the physical world, dialogue ambiguity creates significant communication challenges [10, 11]. Text-based instructions struggle to convey spatial relationships effectively, as linguistic descriptions rely on deictic expressions (e.g., “this”, “that”, “here”, “there”) whose meanings depend on shared physical context absent in text-only interfaces [21, 34]. This spatial ambiguity forces users to mentally translate abstract verbal descriptions into concrete spatial representations, increasing cognitive load and potential for misinterpretation [20, 70].

Text-based responses also lack engaging qualities of dynamic presentation formats. Research shows animated and multimedia presentations capture attention better and improve retention compared to static text [42, 48]. Unlike text-only instructions, animations leverage dual-channel processing—combining visual and verbal information to enhance comprehension [54, 62]. This advantage is particularly critical for spatial tasks, where animated demonstrations illustrate movements and spatial relationships far more effectively than textual descriptions [72].

Building on this trajectory, we pursue assistants that not only understand spatial context but also provide embodied, visually grounded guidance, addressing the engagement and clarity gaps of text-only dialogue.

## 2.2 LLM-Powered Assistants in Extended Reality

Extended Reality (XR) represents an emerging technology that bridges the digital and physical worlds by directly overlaying virtual content onto our surrounding physical space, while simultaneously sensing and understanding the physical environment through built-in sensors and cameras. This bidirectional integration enables both the display of virtual information and the comprehension of physical context such as objects, surfaces, and ongoing user operations, making XR particularly well-suited for spatially grounded tasks where digital guidance can be spatially placed and contextually adapted to users.

Researchers have explored the implementations of LLM-powered assistants in XR for various applications. For example, Dogan et al. [12] developed XR-Objects to endow everyday items with “augmented object intelligence,” using real-time recognition and an LLM to surface object-anchored context menus for queries and actions. Lee et al. developed GazePointAR, a context-aware voice assistant that uses eye gaze and pointing to resolve pronouns in object identification queries (e.g., “what’s this?”) using wearable AR glasses [38]. Zhao et al. developed Guided Reality, which combines LLMs and vision models to generate multi-step AR task guidance with dynamic, step-specific visual overlays [74]. The sensing capabilities of XR devices enable these systems to focus on specific applications, e.g., object identification, information retrieval, and procedural task

guidance, which rely on understanding the user’s spatial intent through camera feeds and hand or controller tracking [36]. However, an asymmetry of spatial communication emerges as existing systems mostly rely on text-based responses or virtual UI elements that may introduce extra learning effort for users [1, 8].

The diverse and unpredictable nature of user requests remain a fundamental challenge for general-purpose conversational assistants in XR. Unlike task-specific systems, a truly versatile XR assistant requires a uniform and scalable approach capable of addressing the full spectrum of spatially grounded inquiries. Such a system must effectively communicate various types of spatial information—object locations, directional guidance, action demonstrations, and contextual references to physical entities—while maintaining user engagement. Consequently, we are motivated to empower agents to move beyond pre-scripted behaviors to support open-ended, spatially grounded inquiries.

## 2.3 Hand Gestures in Human–Human Conversations

When people communicate face-to-face, they naturally employ hands to supplement verbal expressions, with research showing gestures play a crucial role in conveying meaning and facilitating comprehension [49].

**Deictic gestures** establish spatial reference by pointing to objects, locations, or directions, enabling joint attention between communicators [33, 41, 67].

**Iconic gestures** represent concrete aspects of the world through form and movement, functioning as visual representations of actions, objects, or spatial properties [49, 51]. These gestures demonstrate how actions are performed, serve as proxies for physical objects by depicting shape or size, or express numeric values through hand configurations [19, 25]. Research shows iconic gestures significantly enhance comprehension, particularly when conveying spatial information and action sequences [24, 31].

**Emblematic and regulatory gestures** encompass culturally specific hand signals carrying conventional meanings, such as thumbs up for approval or thumbs down for disapproval [13, 32]. This category includes comparative gestures where hand height differences illustrate relative magnitudes, and attention-gathering behaviors like repetitive tapping [4, 50]. These conventional gestures function as visual vocabulary that can substitute for or complement verbal messages [18, 55].

**Interpersonal touch and hand-to-hand interactions** serve distinct communicative functions beyond individual gestures. These tactile exchanges fulfill social purposes, such as high-fives for celebrating shared achievements [16, 22], as well as guidance purposes, where physical contact like holding someone’s hand can prevent actions, redirect behavior, or issue warnings [30, 35].

Inspired by these natural hand-based communication patterns, we design AGENTHANDS to generate multiple forms of hand expressions to effectively address spatially grounded user inquiries. This multi-modal approach allows our XR assistant to convey spatial information and guide actions beyond text-only limitations, creating more natural and engaging XR interactions.

## 2.4 Hand-Gesture-Based Embodied Agents

Gestures are fundamental to human communication and have been extensively studied in human-agent interaction across both physical and digital domains. In the physical realm, robotics research has demonstrated that gestures significantly enhance collaboration and narrative delivery. Early works, such as Leonardo [23], utilized predefined deictic and emblematic gestures to facilitate object reference and convey attitude back in 2002. More recently, Huang and Mutlu showed that robots employing deictic, iconic, and beat gestures improve user information recall [27], while Sauppé and Mutlu categorized deictic subtypes (e.g., pointing, presenting) to establish joint attention [61]. Similarly, Salem et al. developed frameworks for synchronizing synthetic speech with gestures to improve social perception [60]. However, physical robots face inherent limitations as general-purpose agents: bulky hardware restricts mobility and ubiquity; rigid mechanics raise safety concerns for close-range interaction; and mechanical movement constraints often hinder natural expressiveness, potentially inducing uncanny valley effects [47, 52].

Conversely, the integration of gesture-enabled agents in the digital domain offers high agility and expressiveness but lacks physical presence. Recent advances in generative AI have enabled high-fidelity, audio-driven gesture synthesis [3, 5] and semantic gesture generation that aligns with conversational content [73]. Systems like Dittos and DialogLab leverage these capabilities to create realistic avatars for virtual telepresence [26, 39]. Yet, despite their visual fidelity, these agents are confined to screens or purely virtual environments. They lack access to the user’s physical reality, limiting their ability to provide spatially grounded assistance such as pointing to real-world objects or demonstrating tasks in the user’s actual environment.

Motivated by the continuous exploration of gestures for embodied agents as reviewed by Wolfert et al. [71], we empower general-purpose LLM agents with gestural capabilities specifically optimized for the XR domain. XR occupies a unique convergence point: it offers the accessibility and agility of digital agent representations, free from the mechanical and safety constraints of robotics, while enabling the spatial interactivity with the physical world that purely virtual agents lack. To navigate this design space, we aim to identify a set of design attributes that systematically characterize how LLM-driven gestures can augment spatial clarity and engagement specifically within the XR realm.

## 3 Formative Study

To identify design concerns for an embodied hand agent in XR, we conducted a two-session formative study combining scenario elicitation with structured brainstorming. Our inquiry centered on two questions:

- **RQ1: Within XR contexts, when are text/speech-only responses sub-optimal, hindering spatial grounding and reducing engagement?**
- **RQ2: What attributes should an embodied agent possess to effectively coordinate timing, representation, and interaction through co-speech hand behavior in XR?**

The study yielded two key outcomes: a set of design attributes defining the hand agent’s behavior, and a novel taxonomy of hand gestures. Together, these frameworks guide the design of hand embodiment to augment conversational agents’ verbal responses in XR.

### 3.1 Participants and Procedure

We recruited 10 company employees (7M, 3F) with diverse expertise in UX/HCI research, XR interaction design, and hand/gesture input. Regarding ethnicity, 3 identified as East Asian, 3 as White, 2 as Black, and 2 as South Asian, which ensured a diverse outcome in terms of cultural contexts. All reported regular use of conversational agents and frequent XR headset use for work or entertainment. None had contributed to the system design prior to the study. We purposefully sampled experienced practitioners rather than novices to elicit domain-informed insights into challenges and design considerations for hand-based embodiment in XR. Detailed participant demographics are provided in Appendix E (Table 1).

Sessions were conducted in two parts (total 90 minutes). We first introduced examples from prior work covering conversational agents, XR guidance systems, and co-speech gesture literature to establish a shared reference frame. In session 1, we asked the participants to identify scenarios where text/speech guidance fails or degrades user experience for spatial queries in XR. In session 2, the participants listed attributes of an effective hand-based embodiment for those scenarios (timing, scale, precision, interactivity, safety signaling, etc.). In each session, we conducted an open discussion to surface trade-offs and tensions. To approximate real-world use, the participants were encouraged to use off-the-shelf conversational agents while uploading photos either captured in situ or sourced from the web.

### 3.2 Results

Two researchers conducted affinity mapping, iteratively re-clustering sticky notes and sketches to validate and refine theme boundaries across sessions. Meanwhile, they mapped emergent themes to constructs in gesture/grounding literature to identify convergences and gaps that informed the resulting design space. We highlight the results from the analysis and brainstorming feedback below, then distill a set of hand agent attributes and a novel taxonomy as a high-level design guideline.

#### 3.2.1 Occasions where verbal/textual responses are sub-optimal.

The participants consistently described breakdowns in purely textual or speech-only guidance when tasks required shared attention to concrete places, parts, or motions. This aligns with prior work showing that gesture supports spatial reference, grounding, and repair in human–human communication [11, 21, 33, 49] and that XR overlays can still impose cognitive mapping costs [1, 8]. Below are the five recurrent occasions that emerged (O1–O5), and we list one to two representative scenarios and user comments for each condition.

**O1: Spatial disambiguation.** The participants described difficulty interpreting location-related instructions from AI, where ambiguity often exists.

*“It says ‘use the oil on the left’, but there are two left of the stove, olive or canola?” (P3)*

*“Put it to the right’ – My right as I sit, or the room’s right? I need the cue from where I’m standing.” (P4)*

**O2: Demonstrations of an action.** The participants found verbal descriptions insufficient to convey subtle body movements. They wanted to see the movement performed rather than infer it from words alone.

*“Flip gently doesn’t tell me how to make it gently. Do I slide under and roll it over, or lift and turn? I need a demonstration.” (P4)*

*“Drive from the hips is vague without the arc—how far back, how high, and what tempo?” (P7)*

**O3: Conceptual representations.** For abstract or unfamiliar concepts such as category names, part labels, or generalized shapes—pure verbal responses are easy to miss and hard to ground. Participants preferred a brief visual/gestural reinforcement (e.g., outlining a shape, indicating a part) to anchor meaning and keep the conversational flow.

*“If the agent says this deburring tool ‘cleans edges’, show a quick sweeping pass along a corner—without that little animation, I don’t really grasp what the tool actually does.” (P4)*

*“When it starts talking about the extruder assembly, I lose it unless you animate what it is—trace its outline, then show a tiny feed motion so I can follow what part does what.” (P3)*

*“When it just says ‘you may get hurt’, I tune it out—but if it can show some bad results, that warning might actually land.” (P7)*

**O4: Precision.** Vague size or angle descriptors require users to mentally estimate the concept, leading to potential misunderstanding. They would appreciate precise visual demonstrations.

*“A handful of nuts’. I need a sense of size so I don’t overshoot calories.” (P5)*

*“‘Tilt slightly’ doesn’t show the angle or the little wrist twist I need.” (P10)*

**O5: Social cues.** Critical information and tone-setting cues are easy to miss in text-only replies. Participants wanted salient attention grabs (e.g., to flag urgency or step changes) and social signals (e.g., to show presence, respect, and turn-taking).

*“Avoid mixing cleaners’ is easy to miss in text—give me a visible cue so I actually pause before I act.” (P2)*

*“When I start talking, a quick wave to get my attention and then ‘listening’ hands make it feel respectful, like you’re with me, not just dumping text.” (P4)*

Across these cases, the participants reported that pure verbal/text responses forced a mental translation from abstract description to situated action, echoing classic observations about the role of deictic and iconic gestures in maintaining common ground [11, 33, 49, 71]. Building on these beginning points, the participants described scenarios that allowed us to distill a set of design attributes for embodied agents exclusive to the XR domain.

**3.2.2 Attributes of embodied agents in XR.** We identified three key attributes (A1-A3) from the participants’ responses that play critical roles for designing embodied agents with hand behaviors in XR.

**A1: Temporal Synchrony.** The participants repeatedly favored tight, synchronized timing between words and hand behavior, describing it as “easier to track” and “more natural”, and noting confusion when gestures lagged or led by more than “a beat”. Their comments clustered around recurring linguistic cues.

- Deictics. *“When it said here/this, the point should land right then. If it comes late, I’m already looking somewhere else.” (P3)*
- Spatial relations. *“Saying above/left of should come with a tiny move that shows that relation at the same moment.” (P7)*
- Action verbs. *“I expect the motion to start on the verb such as rotate, pull, press, not after.” (P3)*
- Quantities. *“Counts or sizes should hit exactly on the number.” (P5)*
- Comparatives. *“When it says bigger than or instead of, I want a flip between the two things right on those words.” (P8)*
- Sequencing. *“Structure words like first/next should get tiny beats, and warning needs the stop cue the moment it’s said.” (P4)*

This preference aligns with foundational gesture research in human-human communication [49] and robotic gesture generation [27, 60], which posits that the “growth point” of a gesture must co-occur with speech prosody to be intelligible.

**A2: Spatial Placement Strategy.** The participants framed “where a response lives” as a choice of reference space, and favored matching that space to the kind of utterance.

- Non-spatial. For general advice or definitions, the participants preferred no anchoring: *“If it’s just context, don’t pin it somewhere random.” (P1)*
- Object-anchored. When naming parts or features, the participants wanted gestures/overlays bound to the object’s own frame: *“Say the fan intake, show it on the printer itself, not the room.” (P4)*
- User-relative. For immediate actions (“grab”, “place”, “turn left”), the participants preferred cues referenced to their current pose and reach: *“Tell me left/right as I see it, within arm’s length.” (P6)*
- World-anchored. For navigation and multi-step setups, participants valued anchors tied to stable room features (desk edge, wall display): *“Keep the next step where I’ll walk to—it shouldn’t drift.” (P6)*
- Path/trajectory space. For verbs like rotate/trace/route, participants wanted guidance expressed as paths with start/end keyed to the chosen frame: *“Show the move in the space where it actually happens. For example, arc over the knob, route along the panel.” (P9)*
- Selection/comparison space. When contrasting options, they preferred a neutral “comparison band” that spans referents: *“Line them up in one strip so the difference is obvious.” (P7)*

We identify that users may expect XR agents to discard the kinematic constraints of the physical world. Unlike robots, which must navigate around obstacles, or humans, who are bound by gravity, participants expected the agent to exhibit *freeform agility*—passing through solid tables to reach a target, hovering in mid-air for better visibility, or teleporting instantly to a point of interest. As P3 explicitly noted: *“I don’t need it to walk around the table like a person; just blink and be at the nozzle.”* This suggests that in XR, adhering to realistic physics (e.g., walking time, collision) is often perceived as friction; users prioritize *functional ubiquity*, where the agent is unconstrained by physics and simply exists wherever it is needed, instantly. Furthermore, users expressed a desire for

*intimate proximity* impossible for physical beings; they wanted the agent to float directly in their personal space during difficult steps, acting as an omnipresent guide: “*Sometimes I want it floating right in front of me, like a close friend whispering the answer, rather than shouting from across the room.*” (P7) This suggests that in XR, functional ubiquity—being exactly where the user needs to look, regardless of gravity or personal space norms—supersedes realistic positioning.

**A3: Gesture Intents.** The participants’ feedback suggests that hand behaviors serve multiple purposes beyond pointing, and that expressiveness should be functional—advancing grounding, flow, and engagement rather than ornamental. We summarize the purposes into the following complementary dimensions.

- **Spatial referencing.** Establish and maintain common ground about where, what, and how much. “*When you say this part, point right at it so I don’t scan three look-alikes.*” (P8), “*If it’s 30 cm from the window, show that distance in place so I know when to stop.*” (P7)
- **Demonstration.** Act as tools or act out actions to reveal affordances. “*Don’t just say rotate; show the wrist twist and how far the knob should turn*” (P10), “*Mimic the scraper so I see the angle to lift the print without gouging it.*” (P3)
- **Didactic structuring.** Scaffold learning by sequencing and pacing instruction. “*Mark first/next steps with the corresponding gestures so I follow the step change.*” (P5), “*Give me a quick 3-2-1 before the tricky part so I’m ready.*” (P7)
- **Interactive engagement.** Turn monologue into dialogue via lightweight, situated controls. “*If you’re unsure, show two spots and let me tap the right one instead of typing.*” (P9), “*Pop small options by the hand for me to either confirm or to try alternative.*” (P2)
- **Affective expression.** Convey beauty or aversion to frame meaning and mood. “*When you highlight a nice detail, add a gentle flourish so it feels special.*” (P1), “*If something’s gross or unsafe, a tiny recoil makes me hesitate in a good way.*” (P10)
- **Social signaling.** Modulate tone, trust, and motivation through subtle, legible cues similar to real-person communication. “*Thinking gestures to let me know you are paying attention or thinking about your answers.*” (P3), “*For warning, show the stop cue exactly when you say it so I know you really care about me.*” (P7)

These findings highlight two new design territories unexplored by prior art in robotics or humanoid avatars. First, *semantic materiality*: unlike physical robots restricted to fixed hardware, participants expected the XR agent to utilize visual effects as a communicative channel—glowing to warn of danger or becoming transparent to show occluded parts. Second, *tangible interaction*: users expressed a desire to physically interact with the agent (e.g., high-fiving, grabbing the hand to pause). This implies that effective XR agent design must extend beyond kinematic motion to include dynamic visual properties and direct, touch-based interactivity.

**3.2.3 Taxonomy of the hand gestures.** Synthesizing the formative results and insights, both the occasions where verbal responses are insufficient (O1–O5) and the attributes designers must consider (specifically A2–A3, where A1 will be addressed in the system design), we distill a set of core design dimensions with representative elements as a comprehensive taxonomy for embodied hand-agent design (Fig. 2):

**Handedness:** An intrinsic choice of single vs. two-hand behaviors. Two hands enable comparative and bi-manual cues; a single hand reduces visual load.

**Gesture:** A set of legible hand gestures. It addresses the occasions where pure text/verbal are not enough. For instance, a palm gesture can express a stopping intention (O5), and a hold gesture can serve as object shape (O3).

**Spatiality:** Gestures vary by reference space and proximity, addressing spatial match (A2). Mid-air works well for demonstration and expressiveness (O2, O5); object-anchored gestures resolve disambiguation (O1); object-region (part-specific) supports precision and conceptual labeling (O4, O3); user-relative placements support social cues with engaging interactions (O5).

**Temporal dynamics:** Not all gestures should be static. Animated variants convey process and function. For instance, a rotation to demonstrate motion (O2), a pour to depict tool function (O3), or a wave to attract attention (O5), while static poses serve succinct cues and rests.

**Interactivity:** Beyond standalone motions, gestures can invite micro-interaction (e.g., high-five, handshake), enriching social cues and engagement (O5).

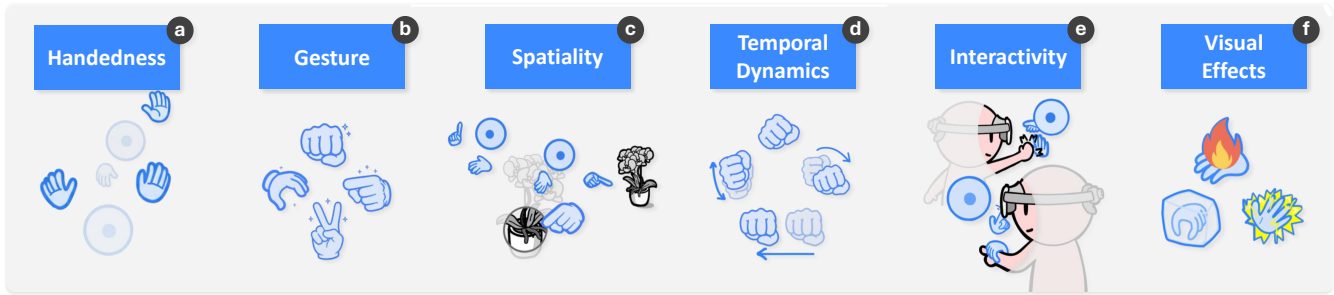
**Visual effects:** Subtle visual effects (e.g., heat/burn, freeze, glow) heighten salience (O5) and make conceptual representation more graspable (O3) without overloading the scene. Research also suggests that observing affected body parts echoes the feeling of first-hand experience [17, 64], hence warnings may be more viscerally salient when overlaid on the agent’s hand.

These dimensions provide an open-ended design guideline for co-speech, embodied agent design in XR domain. Each gesture behavior can be defined by selecting values along each dimension.

In summary, the proposed taxonomy is compositional and readily scalable: designers can introduce new combinations as needed while staying within its structure. In the following sections, we introduce the design of AGENTHANDS with a library of hand gesture behaviors and evaluate that this taxonomy effectively guides the creation of a usable, spatially sensitive hand embodiment for XR conversational agents.

### 3.3 Embodiment of the hand agent in XR

To effectively enact the taxonomy and the key attributes, we derived a specific embodiment strategy. We represent the XR agent as two semi-transparent hands and a small glowing orb (Fig. 1). While prior works have explored hand gesture synthesis primarily within photorealistic, full-body avatars to enhance social realism (e.g., [39, 71]), we deliberately opted for a minimalist abstraction. Richer embodiments with faces and bodies introduce additional social cues and anthropomorphism that can influence perceived competence and affect, making it harder to attribute observed effects specifically to hand gestures. This also aligns with recent findings suggesting that simpler agent representations can reduce human cognitive load while maintaining instructional effectiveness [66]. To this end, the hands are rendered as transparent blue meshes with light particle effects. This keeps the agent recognizable as “hands” while remaining neutral with respect to age, gender, and ethnicity. The glowing orb serves as a locus of attention that indicates where the agent is “focusing” without invoking facial expressions or gaze



**Figure 2: A taxonomy of the hand gestures, distilled from our formative study and defined by six key dimensions: (a) *Handedness* (single, two-hand). (b) *Gesture* (e.g., palm, fist, point, V-sign). (c) *Spatiality* (mid-air, object-anchored, object-region). (d) *Temporal Dynamics* (e.g., static, rotation, shaking, pouring). (e) *Interactivity* (non-interactable, high-five, hold/shake hand). (f) *Visual Effects* (e.g., default, heat/burn, freeze, glow).**

behavior. This minimalist form factor also addresses the *physics-defying agility* identified in A2. In XR domains, a full-body avatar would either occlude the workspace or need to stand farther away, reducing spatial clarity. Our design allows the agent to “reach into” the workspace with precise, low-occlusion guidance, foregrounding gestures while preserving visibility of the environments and the objects.

## 4 AGENTHANDS

### 4.1 System Walk-through

Fig. 4 presents the AGENTHANDS pipeline for producing and rendering spatially grounded responses with a hand agent in XR.

Before the conversational session begins, AGENTHANDS performs a lightweight pre-registration of the environment (Fig. 3). The XR headset wearer needs to look at and confirm the objects that should be interactable (e.g., plants, tools, furniture). For each object, we store a semantic label and a 3D bounding box in an object registry.

To start a conversational session, the XR headset wearer initiates a session by asking about a plant in the real world (Fig. 4a). The utterance is captured via a speech-to-text (STT) module (Fig. 4b). This utterance, together with the first-person view (FPV) and gaze ray (Fig. 4c), and a pre-registered object registry (Fig. 4d), are streamed to the back-end LLM engine.

Conditioned by a response generation system prompt (Fig. 4e; see Appendix B), the LLM produces a conventional textual response augmented with inline *GestureEvents* (Fig. 4g). Each *GestureEvent* attaches to specific triggering words and encodes the primitives for a hand behavior following the dimensions in the taxonomy (Fig. 4f).

The structured output returns to the XR headset, where a lightweight planner parses the inline *GestureEvents*, resolves object references against the object registry, maps anchors to world coordinates, and parameterizes the corresponding hand motions. The textual reply is synthesized via a text-to-speech (TTS) module (Fig. 4h), and word-level timestamps (Fig. 4i) are used by a scheduler to align each parsed *GestureEvent* to the onset and span of its trigger word(s).

In the illustrated example, the agent says: “You can use *this* pot to water the orchid. Gently *pour* some water onto the soil.”

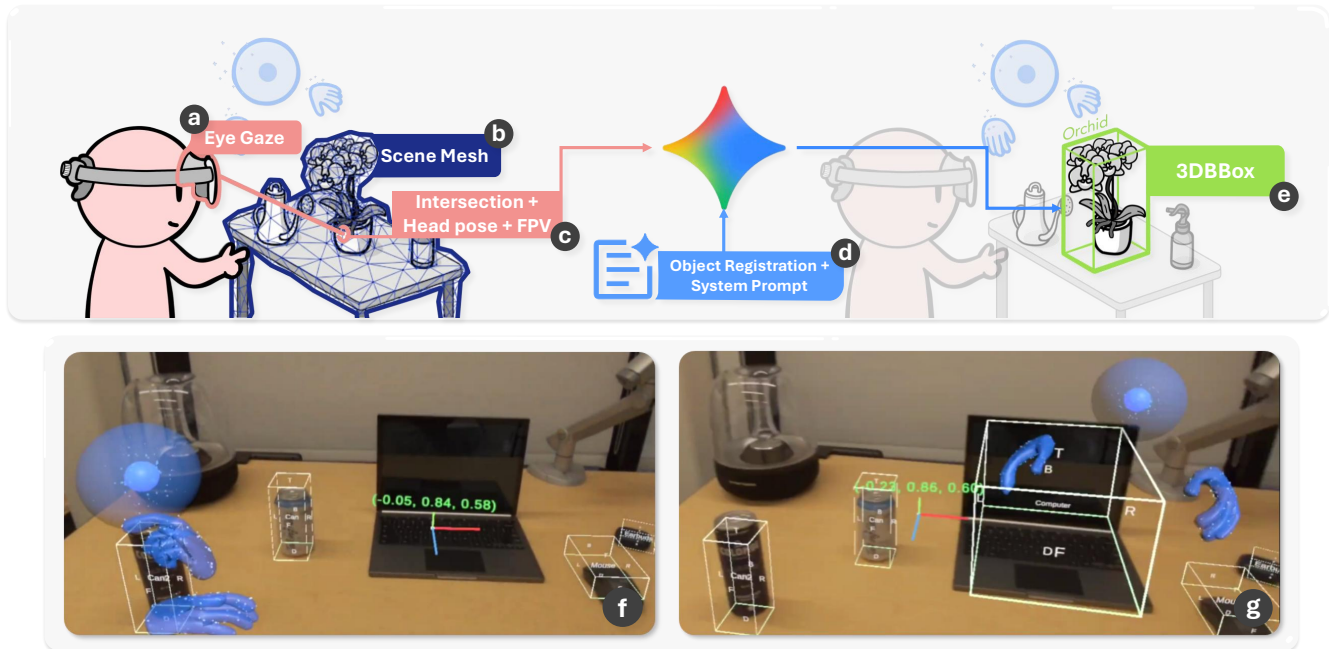
The visualized hand points to the watering pot on “*this*”, then performs a pouring motion on “*pour*”, yielding synchronized, spatially grounded guidance.

### 4.2 Object Registration in the Object Registration Mode

To realize the rich, spatially sensitive hand gestures identified in our taxonomy, the agent must reliably understand the physical scene. Given the computational cost and current accuracy limits of fully on-device 3D object tracking, we adopt a lightweight pipeline that fuses eye gaze, device-side depth reconstruction, and a multi-modal LLM to semi-automate registration of objects in the user’s first-person view (FPV) (Fig. 3). This design choice preserves our paper’s core contribution, while enabling a smooth end-to-end experience under realistic resource constraints. We later discuss limitations of this approach in the Limitations section.

The goal is to register two pieces of key information about each object: (1) the label, and (2) the spatial properties (position, rotation, and dimensions represented as a 3D bounding box) needed to place the agent’s hands next to the target object. Meanwhile, to support a higher-granularity object-region *Spatiality* dimension, we post-process the oriented bounding boxes to generate six faces, and nine regions for each face.

We introduce the workflow to register an object in Fig. 3. At any stage of operating AGENTHANDS, a user can enter the object registration mode by simply talking to the agent to register new objects. When the user says “register this”, the system casts an eye gaze ray from the headset position (Fig. 3a) onto a scene mesh constructed by a scene understanding module (Fig. 3b) to get a 3D intersection point. At the same time, first-person-view RGB images are captured at a fixed frequency (Fig. 3c). We project the intersection point onto the image to crop the image with an empirically set size, and send the image together with the headset pose, and intersection position to a back-end LLM processor. Instructed by an object registration system prompt (Fig. 3d; see supplementary material), the model predicts (1) the semantic label of the object, (2) the estimated 3D dimensions of the object, and (3) the estimated orientation of the object (forward points to the headset while upward points to global



**Figure 3: Top: Object registration pipeline.** (a) The XR headset tracks the eye gaze, while (b) an on-device scene understanding module reconstructs a scene mesh. (c) The 3D gaze ray intersects the mesh to yield a target point; this point, together with head pose and the egocentric first-person view are sent to the back-end LLM engine. (d) An object registration system prompt instructs the model to infer the target object’s data including the label, the transform represented by an oriented 3D bounding box. (e) The output data is sent back to the XR headset, and the object is registered in the object registry. **Bottom: Real-world deployment of the object registry process.** The user first speaks “enter object registration mode”. Then, (f) the user looks at the laptop and says “register”. (g) AGENTHANDS registers the laptop following the pipeline illustrated above.

up if unable to estimate). The data is sent back to the XR headset. Using the orientation data, the system also assigns the labels of the six faces (front (F), back (B), top (T), bottom/down (D), left (L), and right (R)) and nine regions for each face (vertical: bottom-middle-top  $\times$  horizontal: front-center-back pairs) to the bounding box. Finally, AGENTHANDS adds the data to an object registry to complete the registration of one object. To inform the user of the registration, the system visualizes an oriented 3D bounding box with the given label, size, and orientation (Fig. 3g), visible only in the object registration mode. Once an object is registered, the user can continue registering more objects or exit the mode by verbal request.

The object registry will be used as the spatial reference of the hand gesture behaviors with objects’ region-level awareness (e.g., bottom of a plant, right side of a coffee machine). The spatiality is encoded in the format: *id\_face\_verticalRegion\_horizontalRegion*. For instance, *orchid\_R\_bottom\_center* means the bottom-center region on the right side of the orchid.

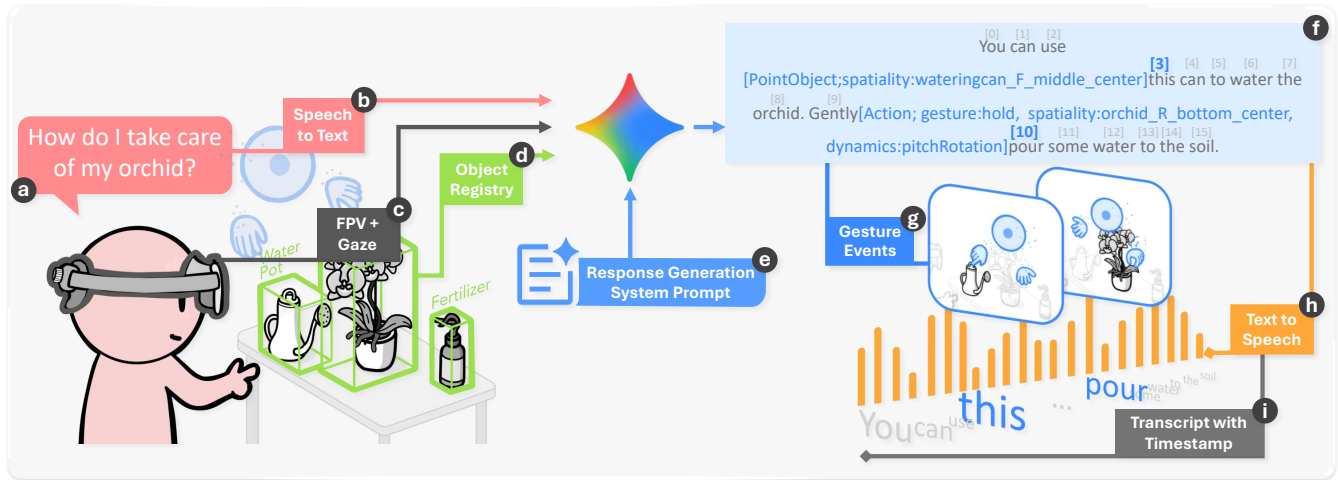
### 4.3 Hand Gesture Library

By selecting a combination of the elements across all the dimensions from the taxonomy (Fig. 2), we construct a **Hand Gesture Library** (Fig. 5) that serves as a representative set of hand gesture behaviors useful for XR conversational AI interactions. For

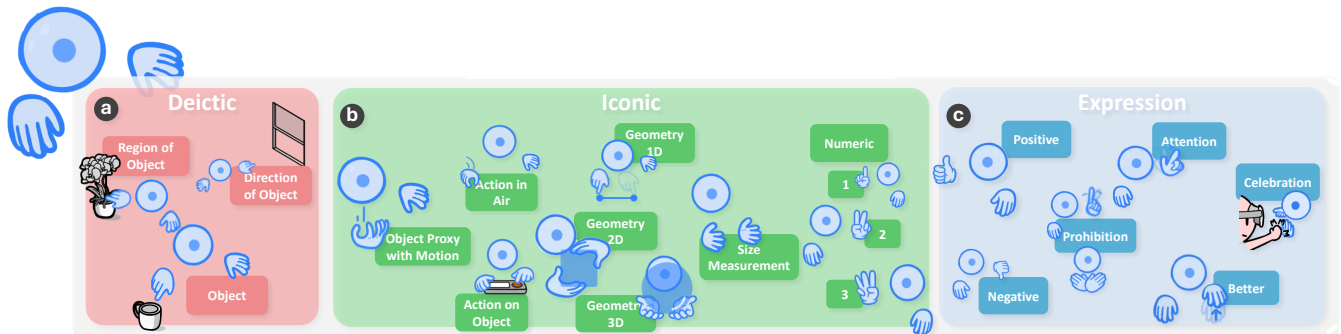
instance, by combining *Handedness*: single-handed, *Gesture*: pointing gesture, *Spatiality*: object-anchored, *Temporal Dynamics*: static, *Interactivity*: non-interactable, and *Visual Effects*: default, we get a *PointObject* gesture. Similarly, by combining *Handedness*: double-handed, *Gesture*: palm gesture, *Spatiality*: object-region, *Temporal Dynamics*: static, *Interactivity*: non-interactable, and *Visual Effects*: default, we get a *SizeMeasurement* gesture. To support dynamic mimicry, combining *Handedness*: single-handed, *Gesture*: pointing, *Spatiality*: object-anchored (path), *Temporal Dynamics*: animated (tracing), and *Visual Effects*: default yields a *TraceTrajectory* gesture (e.g., routing a cable). Likewise, for safety signaling, combining *Handedness*: single-handed, *Gesture*: palm (stop), *Spatiality*: object-anchored, *Temporal Dynamics*: animated (recoil), and *Visual Effects*: red glow/smoke produces a *HeatWarning* gesture that viscerally communicates hazard.

To address the gesture intents (A3), we further group these gestures by their roles in the conversational interaction with the agent. Specifically, we propose the following categories and use them in the response generation system prompt (Fig. 4e) to facilitate the gesture selection based on the semantics of the textual responses.

- **Deictic**: Referencing entities and locations. It supports spatial disambiguation and attention (O1, O5).
- **Iconic**: Depicting concepts/objects and actions. It supports conceptual representation and demonstration (O3, O2).



**Figure 4: Workflow of AGENTHANDS after object registration.** (a) A user asks “How do I take care of my orchid?”. This sends three data streams to the back-end LLM, including: (b) the transcribed speech-to-text (STT) query, (c) the egocentric first-person view (FPV) with gaze point, and (d) the pre-registered scene object registry. We provide (e) a system prompt instructs the LLM to generate: (f) a structured response with inline co-speech *GestureEvents* that specifies (g) gesture type and necessary parameters. At runtime, (h) the response is synthesized to speech via a text-to-speech (TTS) module, and the *GestureEvents* are scheduled to animate following the (i) word-level timestamps.



**Figure 5: Hand Gesture Library of AGENTHANDS, enabling expressive and interactive co-speech behaviors in XR conversations.** We group them by the semantic purposes that can facilitate the LLM to understand which gesture instantiation to select, (a) deictic, (b) iconic, and (c) expression.

- Expression: Conveying emotion, support, and aesthetic emphasis. It strengthens social cues and engagement (O5).

Our scenarios provide broad coverage of taxonomy-guided applications, and additional element combinations can be explored; limitations and extensions are discussed in Section 7.

We formalize the definition of each gesture using a template output structure, *GestureEvent*. Each *GestureEvent* is bound to a triggering keyword in the response and use such binding to achieve the temporal match raised in the formative study (A1). In general, a *GestureEvent* strictly follows the format:

[id; attribute<sub>1</sub>:value<sub>1</sub>, attribute<sub>2</sub>:value<sub>2</sub> . . . ]keyword

The *id* is the unique identification of the supported hand gesture listed in Fig. 5. The *attribute<sub>i</sub>:value<sub>i</sub>* pairs are the parameters needed for the corresponding gesture with the supported values mentioned

in Fig. 2. To reduce the token size of the system prompt, we set a default value for each attribute and omit attributes for gesture instantiations using the default value in the *GestureEvent*. Lastly, the *keyword* is the triggering word of the gesture. Note that for the scenarios where the gesture is more relevant to a short phrase or sentence but not a single word, we instruct the model to attach the *GestureEvent* to the word that has the highest semantic binding, which aligns with the temporal match concern (A1) from the formative study<sup>1</sup>.

Below, we list several example *GestureEvents* bound in the textual responses, please refer to the supplementary material for more gesture instantiations.

<sup>1</sup>refer to Appendix B for the response generation system prompt

“You can use [PointObject; spatiality:wateringcan\_F\_middle\_center]this watering can to water the orchid. Gently [Action; gesture:hold, spatiality:orchid\_R\_bottom\_center, dynamics:pitchRotation]pour some water to the soil.”

“If the wok is on [BurnEffect; spatiality:wok\_F\_top\_center]fire, do [StopExpression]not panic. Grab the lid and quickly [Action; gesture:grab, spatiality:wok\_F\_top\_center, dynamics:zTranslation]cover on the wok. You will be [ThumbupExpression]fine!”

#### 4.4 Runtime Execution in the Conversation Mode

When a user wants to query the agent, the user enters the conversation mode by verbal prompting. AGENTHANDS places the agent hands in front of the user’s view while doing a default thinking gesture indicating waiting for the user input. After the user initiates a prompt, the system sends (a) the transcribed user query, (b) the first-person view and gaze intersection point, and (c) the current object registry to the back-end LLM engine (Fig. 4a-e). Instructed by the response generation system prompt, the augmented response (Fig. 4f) is sent back to the XR headset.

AGENTHANDS performs the following three steps to ensure a response is correctly delivered to the user with both the live speech playback and the synchronized hand behaviors.

**4.4.1 Response parsing.** AGENTHANDS provides a response parser to construct the temporal aligned dictionary from a raw response for the co-speech hand gesture interaction. In the following example:

“You can use [PointObject; spatiality:wateringcan\_F\_middle\_center]this watering can to water the orchid. Gently [Action; gesture:hold, spatiality:orchid\_R\_bottom\_center, dynamics:pitchRotation]pour some water to the soil.”

The response parser fetches the *GestureEvents* as the values of the dictionary, and sets the positional index of the *keywords* as the keys. Specifically, 3: “PointObject; spatiality:wateringcan\_F\_middle\_center” and 11: “Action; gesture:hold, spatiality:orchid\_R\_bottom\_center, dynamics:pitchRotation”.

**4.4.2 Word-level hand gesture alignment with speech playback.** To display synchronized hand behaviors alongside the TTS audio output, we need to know which word is being played at any given time so that we can query the dictionary to trigger the corresponding *GestureEvent*. We compute a smoothed root-mean-square (RMS) energy profile over the audio to reveal speech/silence structure. The transcript is tokenized (punctuation retained). Initial per-word durations are assigned using simple cues—word length, punctuation (short pauses), and basic context (event words slightly faster), and normalized to span the clip with small inter-word gaps. A single forward pass accumulates word start/end times. Then, each boundary between words is nudged toward the nearest local minimum in the energy profile within a short window, improving pause placement and reducing overlap. The result is a word-level timeline (start, end, optional confidence, energy) that is used at speech playback to determine which word is being played.

**4.4.3 Gesture state transitions and interactive hand gestures.** We control each hand with a finite state machine. A hand’s state comprises three independent sub-states: (1) transform, (2) gesture, (3) vfx (visual effect). The initial idle state is a rest pose with no local

gesture or visual effect. When an incoming *GestureEvent* is triggered, if the *Handedness* is single, we first check if there is a hand that is in the idle state, and if so, we trigger the state transition, with the gesture function. Otherwise, assign the event to the hand whose last gesture transition is oldest. If the incoming *GestureEvent* is a double-handed one, we directly apply the gesture state transition to both hands simultaneously. If there is no incoming new *GestureEvents* after a time threshold, we transfer the hand gesture back to an idle gesture state.

AGENTHANDS supports a type of hand gestures that involve the user’s interaction (e.g., high-five, hand shake). We implement asynchronous functions to check whether predefined criteria are met. Using the high-five gesture as an example, we check if collision happens between the agent hand and the user’s hand. Once collide, we trigger the completion of the local high-five gesture. If the criteria are not met after a time threshold, we transition the hand gesture state back to the default, and wait for the upcoming gestures.

#### 4.5 Implementation

AGENTHANDS is implemented using Unity 6000.1.5f1<sup>2</sup>. The STT module parses verbal user queries using the Android SpeechRecognizer API<sup>3</sup>. Agent responses are generated using *gemini-2.5-flash-exp*<sup>4</sup> by sending web requests to the HTTP API endpoint. The TTS module announces agent responses using *gemini-2.5-flash-preview-tts*<sup>5</sup>. User gaze, hand inputs, and scene mesh are available to AGENTHANDS through Unity OpenXR extensions<sup>6</sup>. For our user study, we deploy AGENTHANDS to a Samsung Galaxy XR headset<sup>7</sup>. By default, we use the speech-only interface with captions available for users to toggle on. When enabled, the captions are displayed at the bottom of the view.

### 5 Application Scenarios

We present application scenarios illustrating how AGENTHANDS extends conversational guidance into spatial action across XR contexts:

**Engaging Design Spotlight (Fig. 6a).** The hand agent acts as a product presenter highlighting form and aesthetics in situ. When a user requests a speaker introduction, the agent travels beside it and uses both hands to depict the cylindrical body and spherical top while describing how these shapes create a “stylish look.” The gestures emphasize key design elements, helping users connect abstract design language (e.g., “cylindrical,” “spherical”) with concrete visual features.

**Supporting Decision-Making (Fig. 6b).** When comparing alternatives, the hand agent uses expressive, contrastive gestures to scaffold decisions. In beverage selection, hands hover above coffee and soda cans as the agent narrates differences. As the agent explains coffee is better for fighting tiredness, the hand above it rises

<sup>2</sup>Unity: <https://unity.com/releases/editor/whats-new/6000.1.5f1>

<sup>3</sup>SpeechRecognizer: <https://developer.android.com/reference/android/speech/SpeechRecognizer>

<sup>4</sup>Gemini 2.5 Flash: <https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash>

<sup>5</sup>Gemini 2.5 Flash TTS: <https://ai.google.dev/gemini-api/docs/speech-generation>

<sup>6</sup>Samsung Galaxy XR: <https://www.samsung.com/us/xr/galaxy-xr/galaxy-xr/>

<sup>7</sup>Unity OpenXR: <https://docs.unity3d.com/Packages/com.unity.xr.openxr@1.15/manual/index.html>



**Figure 6: Application scenarios of AGENTHANDS: a) design spotlight description for a speaker, b) drink recommendation for energy boost, c) orchid care tip, d) in-situ 3D printer tutorial, and e) lifestyle companion.**

in a subtle “lift” gesture while the other remains lower, visually encoding the recommendation. This comparative choreography lets users see reasoning unfold directly over options.

**Providing Tips (Fig. 6c).** AGENTHANDS delivers quick, actionable tips grounded in the surrounding environment. For orchid care, the agent appears near the plant and fertilizer bottle, performing a squeeze gesture that mimics applying fertilizer while explaining correct usage. A brief numerical gesture reinforces monthly application. These gesture-accompanied advisories act like spatial sticky notes, encoding “what to do” and “how often” directly onto corresponding objects.

**In-Situ Tutorials (Fig. 6d).** Delivering step-by-step guidance where actions occur, synchronized with the user’s XR pace, showcases our system’s spatial awareness. During 3D-printer onboarding, the agent anchors to the extruder, points to the release lever, and performs an iconic insertion showing angle and depth. A brief 3-2-1 beat precedes heating; warning tint marks the hot end. For calibration, hands trace the bed-leveling pattern and display an in-plane ruler for nozzle distance. When ambiguity arises (“which SD card port?”), candidates are highlighted and the user taps the hand to confirm.

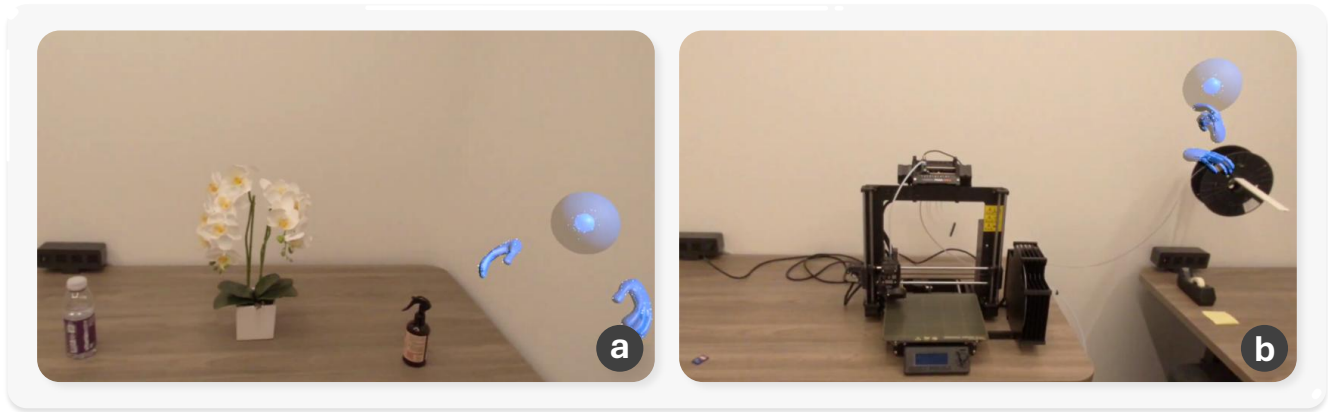
**Lifestyle Companion (Fig. 6e).** The agent serves as a gentle lifestyle coach. When the user asks about an alcoholic drink, the hands perform a “prohibit” gesture while holding the user’s hand, then a thumb-down with warning VFX while explaining excessive drinking is unhealthy and potentially toxic. These socially expressive, hand-only gestures let the agent deliver sensitive advice more embodied and situated than text or audio alone.

## 6 User Study

To evaluate our co-speech gesture generation and visualization system and ensure our design addresses challenges from the formative study, we conducted a within-subjects comparative study with two conditions: (a) a baseline embodied conversational AI agent sharing AGENTHANDS’s interface but without hand visualization, and (b) AGENTHANDS with full co-speech gesture generation. This contrast isolates the value of adding embodiment to an otherwise identical agent and reflects how AGENTHANDS is experienced as a coherent whole. It demonstrates our taxonomy’s effectiveness and the usability of our augmented response generation, parsing, and execution pipeline. We selected two tasks designed to exercise spatial reasoning and multi-step guidance: (a) orchid-care tutorial and (b) 3D-printer operation walkthrough. These domains balance everyday and technical scenarios, require sequential actions, and include spatially sensitive references (e.g., identifying leaves or roots, locating printer components), probing both coarse and fine-grained pointing, object referencing, and safety-related cues. Both conditions used identical researcher-scripted verbal content from the baseline AI agent, ensuring participants received the same instructions and dialogue flow, isolating gesture effects on communication effectiveness and engagement in XR.

### 6.1 Participants and Procedure

We recruited 12 participants from an anonymized company (software engineers, UX designers, UX researchers, and HCI researchers). None had participated in the formative study. Unlike the formative study, this evaluation focuses on AGENTHANDS’s usability for novice users. We imposed no prerequisites on prior AI agent or XR experience. Participants were expected to learn and enjoy the system with minimal onboarding. Mean age was 29.7 years ( $SD = 5.31$ ); 10 identified as male (83.3%), 1 as female (8.3%), and 1 as non-binary (8.3%). *XR experience*: 6 had built XR applications (50.0%), 5 had



**Figure 7: User study setup. (a) Orchid plant care task where users learned about their newly purchased plant and its care. (b) 3D printer tutorial task where users learned printer operation from the embodied agent.**

used XR applications (41.7%), and 1 had never used XR (8.3%). *LLM experience*: 2 had built LLM applications (16.7%), 9 reported frequent use (75.0%), and 1 reported occasional use (8.3%). Additionally, 5 participants (41.7%) reported using LLM assistants for spatially grounded daily tasks. Detailed demographics are in Appendix E (Table 2).

After the demographic survey, participants underwent a 5-minute tutorial on voice-based communication with the XR agent. Participants then completed two procedural tasks with XR agent assistance—one with the baseline agent and one with AGENTHANDS. Participants were told only that they would experience two different designs; condition order was counterbalanced to mitigate novelty and order effects. Each task had 10 steps across three phases: background introduction, main tutorial, and completion confirmation. The detailed tasks and transcripts are attached to the Appendix D.

**Task 1: Orchid care.** This task (Fig. 7a) simulates bringing a new orchid home and seeking care guidance. The agent introduces each orchid component, covering location, name, visual characteristics, and biological function. The tutorial covers three care activities. First, proper watering: frequency, volume, pouring locations, and drainage importance. Second, fertilization: application frequency, mixing techniques (shaking thoroughly), and targeted application to roots and leaves with appropriate dosage. Finally, optimal placement: near a window with indirect sunlight while maintaining distance to prevent direct light damage. Upon completion, the agent provides celebratory feedback.

**Task 2: 3D printer operation.** This task (Fig. 7b) simulates acquiring a 3D printer and learning to print from an SD card. The agent introduces key components—print bed, nozzle, filament, and control panel—explaining each component’s function and location. The tutorial covers the complete printing workflow: powering on and inserting the SD card, navigating the file browser and selecting files using the control knob, and initiating the print process with proper startup procedures and safety considerations like heated bed temperature awareness. Upon completion, the agent provides positive feedback.

**Post-session surveys.** After each task, participants completed surveys with Likert-scale questions (1 = strongly disagree, 7 = strongly agree) evaluating system features and user experience, plus a standard System Usability Scale (SUS) questionnaire [7].

**Open-ended exploration and interview.** After both tasks, participants freely explored AGENTHANDS’s capabilities by posing environment-related questions and observing gesture-based responses to various conversational contexts. Each participant then completed an interview about their qualitative experience with both conditions and thoughts on AGENTHANDS’s pros and cons.

## 6.2 Results

The Likert-scale ratings of specific features and overall user experience for the agent’s responses in both conditions are shown in Fig. 8. The Shapiro–Wilk normality test indicated that the normality assumption was not met. Therefore, we used a Wilcoxon signed-rank test to assess statistical significance.

The participants demonstrated significantly improved spatial understanding (O1, O4) when using AGENTHANDS compared with the baseline, highlighting the effectiveness of our hand spatial placement design (A2). The participants found it easier to locate an object with AGENTHANDS (Q1:  $M = 6.50, p < 0.05$ ) and to identify directions referred to by the agent (Q2:  $M = 6.33, p < 0.05$ ). “At the beginning I didn’t really know which one was the orchid fertilizer until the agent pointed it out.” (P3). “When the agent asked me to place the orchid by the window using the pointing gesture, I immediately looked toward the window.” (P1). When the agent mentioned a local region or component of an object, the participants preferred responses from AGENTHANDS (Q3:  $M = 6.00, p < 0.05$ ). “It really helped me locate the SD card slot on the 3D printer!” (P7). “It was my first time realizing that the orchid’s roots were sticking out of the planter and breathing!” (P12). These positive effects also reflected our word-level alignment design addressing the temporal synchrony attribute (A1). “When it says this, the point lands right then—there’s no guessing.” (P4) “The little beat gesture told me we’d started a new step.” (P6)

Regarding task execution, the participants also found that the required activities were easier to follow with AGENTHANDS (O2,

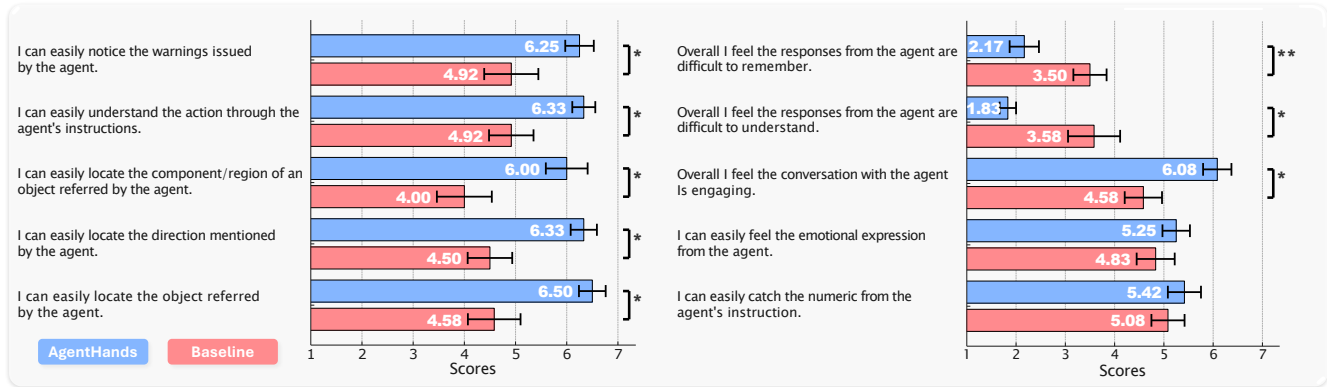


Figure 8: The results of the system feature experience questionnaire. Significance: \*  $p < .05$ , \*\*  $p < .01$ .

O3, O5) (Q4:  $M = 6.33$ ,  $p < 0.05$ ). “Only when the agent showed the lifting gesture did I realize I needed to lift the orchid to let the water drain.” (P5). Warnings issued by the agent were more likely to be noticed in AGENTHANDS (O5) (Q5:  $M = 6.25$ ,  $p < 0.05$ ). “The agent said that if I fertilize too much the roots would burn, and showed a burning visual effect; that was really impressive, and I definitely won’t apply too much fertilizer to the orchid.” (P5).

AGENTHANDS also demonstrated better overall user experience. Compared with the baseline condition, the participants found conversations with AGENTHANDS more engaging (O5) (Q8:  $M = 6.08$ ,  $p < 0.05$ ), its responses easier to understand (Q9:  $M = 1.83$ ,  $p < 0.05$ ), and its instructions easier to remember (Q10:  $M = 2.17$ ,  $p < 0.01$ ). Participants attributed these gains to the timely, unobtrusive delivery of gestures that made the exchange feel natural and seamless: “It’s like the hands were part of the sentence; I didn’t have to stop and think.” (P6), “I barely noticed the system, which is good—I just followed along.” (P2) These remarks echo our emphasis on temporal alignment, suggesting that synchronized gestures reinforced comprehension and memory without adding friction.

Despite the generally more favorable feedback for AGENTHANDS, we observed no significant differences between conditions for numerical information in the responses (Q6:  $M = 5.42$ ,  $p > 0.05$ ). As the participants noted: “I feel the numbers are easy enough to remember in voice.” (P6), and “I think the numeric gesture might be helpful, but right now they don’t last long enough for me.” (P2). Going forward, numeric support should be demand-sensitive and more actionable/persistent (e.g., lingered rulers/ticks, tap-to-pin measurements in place, or quick replay on request).

We observed no significant differences regarding emotional expression in the responses (Q7:  $M = 5.25$ ,  $p > 0.05$ ). We noticed that the participants were consistently impressed by the prosody and “active” tone of the shared TTS, which carried much of the effect in both conditions. As a result, gesture cues were perceived as additive but not decisive for this task set. “Both agents are quite emotional, I loved the speech.” (P9) and “The hands add some warmth, but the voice does most of the emotion for me.” (P6). Yet, many noted that interactive moments (e.g., high-fives) felt motivating and relational even if they didn’t shift the rating scale. “The high-five doesn’t change my ‘emotional’ score, but it makes me smile and want to continue.” (P3)

The average SUS score for AgentHands (80.6) was higher than baseline (70.4). A Shapiro–Wilk test confirmed normality, so we used a paired  $t$ -test, which showed no significant difference ( $t = 1.69$ ,  $p = 0.118$ ). These results suggest that AgentHands does not introduce additional usability obstacles while still delivering gains in engagement and spatial clarity.

### 6.3 Discussion

Based on user study feedback, we identify the following insights that open promising directions for future work on embodied agent interaction.

**6.3.1 Interaction mechanisms with XR agents enabled by embodiment.** Our study suggests that adding hands does more than improve clarity; it fundamentally transforms the interaction style from voice-first to a blended, embodied dialogue where speech, gesture, gaze, and brief touch converge. Qualitative traces revealed that while the speech-only baseline involved frequent pauses and verbal confirmations, the embodied condition shifted participants toward succinct, embodied checks, finer-grained follow-ups, and fewer repairs. This shift also surfaced new interaction mechanisms beyond speech. Participants spontaneously treated the agent’s hands as actionable input surfaces, expressing a desire to interrupt or steer the agent via direct manipulation (e.g., “grab the hand to pause it” [P4]) or social connection (e.g., “feels more engaging when we hi-five” [P6]).

This feedback highlights **virtual tactile interaction** as a critical, under-explored modality for embodied agents. In physical Human-Robot Interaction (HRI), touch is often limited by safety concerns (e.g., collision with rigid motors), hardware fragility, and kinematic reachability [59, 63]. Conversely, XR hands sidestep these physical constraints: they can be safely approached, dynamically resized, or instantly repositioned to be within the user’s reach, while still affording rich contact metaphors such as taps or grabs. This finding extends emerging work on overlaying virtual avatars onto physical robots [65], suggesting that purely virtual agents can be designed as *functionally tangible* interfaces that invite direct physical intervention.

These observations hint at a richer protocol in which hands act not only as expressive output but also as input surfaces for

control (pause, resume, confirm). We identify three research opportunities to formalize this: (1) **Bi-directional gestural protocols**: Formalize a compact “interaction alphabet” where hand poses convey system intent and invite unambiguous user responses (e.g., a high-five for confirmation), supporting fast grounding without over-reliance on speech [69]; (2) **Compositional turn-taking**: Investigate how short verbal units and micro-touches can be orchestrated into predictable interaction patterns; and (3) **Generative UX (GenUX)**: Recent systems such as AgentAR[75] and XR Blocks[40] leverage LLMs to synthesize XR interfaces upon the user’s need. Extending this direction, future work could develop task-contingent, ephemeral affordances that appear on the hands only when needed, preserving gesture as the primary channel while providing just-in-time scaffolding.

**6.3.2 Form factor level-of-detail of the embodied agent.** Our findings suggest that for spatially grounded tasks, a hands-only representation occupies a functional sweet spot: it provides sufficient anthropomorphism to establish partnership (“*the hands felt like a partner guiding me*” [P9]) while avoiding the realism costs of full avatars. Participants valued that the floating hands could move fluidly among dispersed work surfaces (printers, shelves) and utilize stylized VFX (e.g., heat auras) without the risk of the uncanny valley. They prioritized precision and clarity over human-level kinematics or facial nuance.

However, we observed that the optimal level of detail is highly context-dependent. As one participant noted, “*If I’m just talking randomly... I feel a more vivid embodiment [is] more appealing.*” This aligns with prior work indicating that high-fidelity avatars are preferred in socially oriented settings to convey rich interpersonal cues; for instance, *Dittos* utilizes realistic avatars to mirror user appearance in telepresence scenarios [39]. Conversely, in task-focused XR, functionality and low occlusion take precedence. *Embodiment* [6] demonstrates that minimal forms can effectively support voice assistance, while Cao et al. found that full-body avatars in AR machine-operation tutorials often become distracting by occluding the workspace [9]. Our results reinforce this distinction, suggesting that minimalist embodiment is superior for hands-on guidance.

Together, these findings frame embodiment not as a fixed choice but as a spectrum of granularity. Designers might dynamically tune the agent’s form—from abstract hands for precision tasks to fuller torsos for social conversation—and leverage non-human capabilities (e.g., morphing into tools, teleporting) to match the spatial demand. This points toward a future of adaptive embodiment, where the agent’s visual representation shifts fluidly based on the momentary context of the interaction.

**6.3.3 Personalization of the embodied agent.** Feedback revealed that for embodied agents, memory is as much spatial and sensorimotor as it is linguistic. Participants implicitly exposed stable habits (e.g., dominant hand, reach comfort zones), environmental regularities (e.g., tool locations), and stylistic preferences (e.g., gesture tempo, tolerance for overlays). When the hands adapted—slowing down near tight clearances, pre-positioning by the user’s typical starting area, or pairing gestures with subtle annotations—users described the guidance as “*made for me*” rather than merely “*made for this task*”. These patterns point toward personalization that

learns a user’s spatial routines and representational preferences, systematically sculpting gesture scale, timing, and density.

Beyond efficiency, such tailoring addresses cultural adaptation: prior work shows that culturally adaptive interfaces significantly improve user performance and satisfaction [58], and that virtual agents benefit from adapting nonverbal behaviors—including gesture and proxemics—to users’ cultural backgrounds (e.g., utilizing alternative affirmatives to the thumbs-up) [14, 46, 57]. Accessibility considerations further motivate personalization, requiring features such as larger motion envelopes for users with limited visual acuity, optimized low-occlusion paths to avoid obstructing critical task areas, or adjusted gesture speeds to accommodate varying motor capabilities.

The broader opportunity is a “living profile” that couples conversational history with a spatial memory graph of the user’s environment and practices. This supports anticipatory hand placement, proactive safety signaling, and teach-once gesture macros that the agent refines over time under explicit user control.

## 7 Limitation And Future Work

While AGENTHANDS demonstrates the potential of embodied conversational agent in XR, several limitations point to fruitful directions for future research.

**Scene Understanding Limitations.** AGENTHANDS currently relies on a lightweight pre-scan to populate an object registry. While this minimizes system latency—a critical priority for maintaining fluid, co-speech gesture dynamics—it currently constrains operation to relatively static environments. However, extending the system with continuous 3D perception is a clear direction for future work, and recent innovations provide excellent templates for this integration. For instance, LLMR [68] demonstrates how a dedicated agent can handle real-time scene queries, while *ImaginateAR* [37] presents a robust pipeline for registering 3D objects through semantic reconstruction. Following these exemplars, future iterations of AGENTHANDS could integrate similar advanced perception modules—or leverage emerging SLAM-based semantic mapping [15]—to transition from pre-registration to fully online, adaptive scene understanding. Beyond scene dynamism, we also aim to enhance spatial granularity. For fine-grained tasks (e.g., circuit assembly) where freehand gestures may lack precision, future work could investigate hybrid strategies that combine generative gestures with target snapping, depth-aware cues, or programmatic action plans. This aligns with recent HCI approaches like *GuidedReality* [74], leveraging LLMs to semanticize scene affordances and anchor gestures to specific, small-scale components automatically.

**Synthesized gestures from the taxonomy.** In AGENTHANDS, we manually curate a finite set of gesture instantiations guided by the taxonomy. Looking ahead, an LLM-aware pipeline could use the taxonomy as a schema to synthesize novel, previously undefined *GestureEvents* on the fly. This would expand coverage beyond hand-authored cases while preserving design intent. This taxonomic approach facilitates both cross-cultural scalability and expanded affective range. Because the framework abstracts semantic intent from specific kinematic instantiations, future systems could dynamically generate culturally congruent variants (e.g., alternative affirmatives) or diverse emotional expressions (e.g., celebratory

flourishes, empathetic pauses) without altering the core system architecture. Such generation would still require a local validator/parser to map events to safe, renderable parameters and to fall back gracefully when an event cannot be interpreted.

**Integration with XR operating systems and applications.** AGENTHANDS currently runs as a standalone agent. Deep integration with XR operating systems and application-level APIs could enable the hands to point to specific sentences in documents, highlight or “press” UI widgets, drag sliders, or annotate digital content with ephemeral markings. Such capabilities would support seamless transitions between conversational guidance and direct manipulation, yielding a more unified interaction paradigm. Systems such as *EmBARDiment* [6] and *Pointer Assistant* [56] have pioneered this direction. We envision a gesture-aware system API surface for agent-initiated actions, and cross-app protocols for passing intent and context, transforming the agent from an isolated assistant into a first-class part of the XR workspace.

**Larger-scale and longitudinal evaluation.** Our study provides initial empirical evidence that AGENTHANDS increases engagement and improves the legibility of spatially grounded conversations compared to a speech-only baseline. However, these findings derive from a controlled, single-session lab study ( $N = 12$ ), which potentially limits broad statistical generalizability. Larger-scale and longitudinal studies are needed to disentangle novelty effects from sustained utility and to examine how user behaviors—such as trust calibration, fatigue, and gesture preferences—evolve as expertise grows. Expanding evaluations to diverse task domains (e.g., education, manufacturing, healthcare) will also be critical for validating the framework’s robustness across varied contexts.

## 8 Conclusion

We introduced AGENTHANDS, an end-to-end system that augments conversational XR agents with hands capable of generating spatialized co-speech gestures in real time. Grounded in a design space distilled from a focused brainstorming exercise, AGENTHANDS synthesizes, stages, and renders gestures that reference objects and locations in the user’s 3D environment while enabling direct interaction. Across application scenarios—design spotlight, XR tutorials, and lifestyle companion—AGENTHANDS enriched agent–user communication by providing a visual reference channel for spatially grounded dialogue. Our comparative study indicates the approach is feasible and promising, with positive results on usability and perceived spatial clarity. This work surfaces open challenges for embodied agents, including adaptive gesture planning under uncertainty, tighter scene understanding integration, gesture personalization, and evaluation at scale. We see AGENTHANDS as a step toward conversational agents that not only speak about the physical world but can act within it, advancing the design of expressive, embodied interfaces for spatial computing.

## References

- [1] Faisal M. Alessa, Mohammed H. Alhaag, Ibrahim M. Al-Harkan, Mohamed Z. Ramadan, and Fahad M. Alqahtani. 2023. A Neurophysiological Evaluation of Cognitive Load during Augmented Reality Interactions in Various Industrial Maintenance and Assembly Tasks. *Sensors* 23, 18 (2023), 7698. doi:10.3390/s23187698
- [2] Inc. Amazon.com. 2025. Amazon Alexa. <https://www.amazon.com/b?ie=UTF8&node=21576558011>. Accessed 2025-09-02.
- [3] Tenglong Ao, Zeyi Zhang, and Libin Liu. 2023. GestureDiffuCLIP: Gesture Diffusion Model with CLIP Latents. *ACM Trans. Graph.* 42, 4, Article 42 (July 2023), 18 pages. doi:10.1145/3592097
- [4] Janet B. Bavelas, Nicole Chovil, Douglas A. Lawrie, and Allan Wade. 1995. Gestures specialized for dialogue. *Personality and Social Psychology Bulletin* 21, 4 (1995), 394–405. doi:10.1177/0146167295214010
- [5] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. 2021. Text2Gestures: A Transformer-Based Network for Generating Emotive Body Gestures for Virtual Agents. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–10. doi:10.1109/VR50410.2021.00037
- [6] Riccardo Bovo, Steven Abreu, Karan Ahuja, Eric J Gonzalez, Li-Te Cheng, and Mar Gonzalez-Franco. 2025. EmBARDiment: an Embodied AI Agent for Productivity in XR. In *2025 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. 708–717. doi:10.1109/VR59515.2025.00093
- [7] John Brooke. 1996. SUS: A “Quick and Dirty” Usability Scale. In *Usability Evaluation in Industry*, Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester, and Ian L. McClelland (Eds.). Taylor & Francis, London, 189–194.
- [8] Josef Buchner, Katja Buntins, and Michael Kerres. 2022. The impact of augmented reality on cognitive load and performance: A systematic review. *Journal of Computer Assisted Learning* 38, 1 (2022), 285–303. doi:10.1111/jcal.12617
- [9] Yuanzhi Cao, Xun Qian, Tianyi Wang, Rachel Lee, Ke Huo, and Karthik Ramani. 2020. An Exploratory Study of Augmented Reality Presence for Tutoring Machine Tasks. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376688
- [10] Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill. 2000. Embodied conversational interface agents. *Commun. ACM* 43, 4 (2000), 70–78. doi:10.1145/332051.332075
- [11] Herbert H. Clark. 2003. Pointing and placing. In *Pointing: Where Language, Culture, and Cognition Meet*, Sotaro Kita (Ed.). Lawrence Erlbaum (now Taylor & Francis), Mahwah, NJ, 243–268. doi:10.4324/9781410607744
- [12] Mustafa Doga Dogan, Eric J Gonzalez, Karan Ahuja, Ruofei Du, Andrea Colaço, Johnny Lee, Mar Gonzalez-Franco, and David Kim. 2024. Augmented Object Intelligence with XR-Objects. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 19, 15 pages. doi:10.1145/3654777.3676379
- [13] Paul Ekman and Wallace V. Friesen. 1969. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Semiotica* 1, 1 (1969), 49–98. doi:10.1515/semi.1969.1.1.49
- [14] Birgit Endrass, Matthias Rehm, and Elisabeth André. 2011. Planning Small Talk behavior with cultural influences for multiagent systems. *Comput. Speech Lang.* 25, 2 (April 2011), 158–174. doi:10.1016/j.csl.2010.04.001
- [15] F. Fei, S. Xian, R. Yang, C. Wu, and X. Lu. 2024. A Wearable Visually Impaired Assistive System Based on Semantic Vision SLAM for Grasping Operation. *Sensors* 24, 11 (2024), 3593. doi:10.3390/s24113593
- [16] Tiffany Field. 2010. Touch for socioemotional and physical well-being: A review. *Developmental Review* 30, 4 (2010), 367–383. doi:10.1016/j.dr.2011.01.001
- [17] David Freedberg and Vittorio Gallese. 2007. Motion, emotion and empathy in esthetic experience. *Trends in Cognitive Sciences* 11, 5 (2007), 197–203.
- [18] Lauren Gawne and Kensy Cooperrider. 2024. Emblems: Meaning at the interface of language and gesture. *Glossa: A Journal of General Linguistics* 9, 1 (2024), 1–39. doi:10.16995/glossa.9705
- [19] Susan Goldin-Meadow. 2003. *Hearing Gesture: How Our Hands Help Us Think*. Harvard University Press, Cambridge, MA.
- [20] Mary Hegarty. 2011. The cognitive science of visual–spatial displays: Implications for design. *Topics in Cognitive Science* 3, 3 (2011), 446–474. doi:10.1111/j.1756-8765.2011.01150.x
- [21] Oliver Herbort and Wilfried Kunde. 2016. Spatial (mis-)interpretation of pointing gestures to distal referents. *Journal of Experimental Psychology: Human Perception and Performance* 42, 1 (2016), 78–89. doi:10.1037/xhp0000126
- [22] Matthew J. Hertenstein, Rachel Holmes, Margaret McCullough, and Dacher Keltner. 2009. The communication of emotion via touch. *Emotion* 9, 4 (2009), 566–573. doi:10.1037/a0016108
- [23] Guy Hoffman and Cynthia Breazeal. 2004. Collaboration in Human-Robot Teams. *Collection of Technical Papers - AIAA 1st Intelligent Systems Technical Conference 2* (09 2004). doi:10.2514/6.2004-6434
- [24] Judith Holler and Geoffrey Beattie. 2003. How iconic gestures and speech interact in the representation of meaning. *Semiotica* 146 (2003), 81–116. doi:10.1515/semi.2003.083
- [25] Autumn B. Hostetter. 2011. When do gestures communicate? A meta-analysis. *Psychological Bulletin* 137, 2 (2011), 297–315. doi:10.1037/a0022128
- [26] Erzhen Hu, Yanhe Chen, Mingyi Li, Vrushank Phadnis, Pingmei Xu, Xun Qian, Alex Olwal, David Kim, Seongkook Heo, and Ruofei Du. 2025. DialogLab: Authoring, Simulating, and Testing Dynamic Group Conversations in Hybrid Human-AI Conversations. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST, 210)*. ACM, 1–20. doi:10.1145/3746059.3747696

- [27] Chien-Ming Huang and Bilge Mutlu. 2013. Modeling and Evaluating Narrative Gestures for Humanlike Robots. In *Proceedings of Robotics: Science and Systems*. Berlin, Germany. doi:10.15607/RSS.2013.IX.026
- [28] Apple Inc. 2025. Home app (Apple Home). <https://www.apple.com/home-app/>. Accessed 2025-09-02.
- [29] Apple Inc. 2025. Siri. <https://www.apple.com/siri/>. Accessed 2025-09-02.
- [30] Stanley E. Jones and A. Elaine Yarbrough. 1985. A naturalistic study of the meanings of touch. *Communication Monographs* 52, 1 (1985), 19–56. doi:10.1080/03637758509376094
- [31] Spencer D. Kelly, Tara McDevitt, and Megan Esch. 2009. Gestures maintain spatial imagery. *Language and Cognitive Processes* 24, 2 (2009), 377–396. doi:10.1080/01690960802365567
- [32] Adam Kendon. 2001. Gesture as communication strategy. *Semiotica* 135, 1/4 (2001), 191–209. doi:10.1515/semi.2001.060
- [33] Sotaro Kita (Ed.). 2003. *Pointing: Where Language, Culture, and Cognition Meet*. Lawrence Erlbaum Associates, Mahwah, NJ. doi:10.4324/9781410607744
- [34] Alfred Kranstedt, Peter Kühnlein, and Ipke Wachsmuth. 2004. Deixis in Multimodal Human Computer Interaction: An Interdisciplinary Approach. In *Gesture-Based Communication in Human-Computer Interaction (GW 2003)*. LNCS, Vol. 2915. Springer, 112–123. doi:10.1007/978-3-540-24598-8\_10
- [35] Michael W. Kraus, Cassey Huang, and Dacher Keltner. 2010. Tactile communication, cooperation, and performance: An ethological study of the NBA. *Emotion* 10, 5 (2010), 745–749. doi:10.1037/a0019382
- [36] Geonsun Lee, Min Xia, Nels Numan, Xun Qian, David Li, Yanhe Chen, Achin Kulshrestha, Ishan Chatterjee, Yinda Zhang, Dinesh Manocha, David Kim, and Ruofei Du. 2025. Sensible Agent: A Framework for Unobtrusive Interaction with Proactive AR Agent. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 22 pages. doi:10.1145/3746059.3747748
- [37] Jaewook Lee, Filippo Aleotti, Diego Mazala, Guillermo Garcia-Hernando, Sara Vicente, Oliver James Johnston, Isabel Kraus-Liang, Jakub Powierza, Donghoon Shin, Jon E. Froehlich, Gabriel Brostow, and Jessica Van Brummelen. 2025. ImaginationAR: AI-Assisted In-Situ Authoring in Augmented Reality. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. ACM, 1–21. doi:10.1145/3746059.3747635
- [38] Jaewook Lee, Jun Wang, Elizabeth Brown, Liam Chu, Sebastian S. Rodriguez, and Jon E. Froehlich. 2024. GazePointAR: A Context-Aware Multimodal Voice Assistant for Pronoun Disambiguation in Wearable Augmented Reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 408, 20 pages. doi:10.1145/3613904.3642230
- [39] Joanne Leong, John Tang, Edward Cutrell, Sasa Junuzovic, Gregory Paul Baribault, and Kori Inkpen. 2024. Dittos: Personalized, Embodied Agents That Participate in Meetings When You Are Unavailable. *Proc. ACM Hum.-Comput. Interact.* 8, CSCW2, Article 494 (Nov. 2024), 28 pages. doi:10.1145/3687033
- [40] David Li, Nels Numan, Xun Qian, Yanhe Chen, Zhongyi Zhou, Evgenii Alekseev, Geonsun Lee, Alex Cooper, Min Xia, Scott Chung, Jeremy Nelson, Xiuxiu Yuan, Jolice Dias, Tim Bettridge, Benjamin Hersch, Michelle Huynh, Konrad Piascik, Ricardo Cabello, David Kim, and Ruofei Du. 2025. XR Blocks: Accelerating Human-centered AI + XR Innovation. In *arxiv*. 9 pages. doi:10.48550/arXiv.2509.25504
- [41] Ulf Liszkowski. 2008. Before L1: A differentiated perspective on infant gestures. *Gesture* 8, 2 (2008), 180–196. doi:10.1075/gest.8.2.04lis
- [42] Chen Liu, Rafael A. Calvo, and Ronny Lim. 2019. Animating student engagement: The impacts of cartoon instructional videos on learning experience. *Research in Learning Technology* 27 (2019). doi:10.25304/rlt.v27.2124
- [43] Google LLC. 2025. Gemini. <https://gemini.google.com/>. Accessed 2025-09-02.
- [44] Google LLC. 2025. Google Assistant. <https://assistant.google.com/>. Accessed 2025-09-02.
- [45] Google LLC. 2025. Google Home. <https://home.google.com/welcome/>. Accessed 2025-09-02.
- [46] Samuel Mascarenhas, Nick Degens, Ana Paiva, Rui Prada, Gert Jan Hofstede, Adrie Beulens, and Ruth Aylett. 2016. Modeling culture in intelligent virtual agents. *Autonomous Agents and Multi-Agent Systems* 30, 5 (Sept. 2016), 931–962. doi:10.1007/s10458-015-9312-6
- [47] Maya B. Mathur and David B. Reichling. 2016. Navigating a Social World with Robot Partners: A Quantitative Cartography of the Uncanny Valley. *Cognition* 146 (2016), 22–32. doi:10.1016/j.cognition.2015.09.008
- [48] Richard E. Mayer (Ed.). 2014. *The Cambridge Handbook of Multimedia Learning* (2 ed.). Cambridge University Press, Cambridge. doi:10.1017/CBO9781139547369
- [49] David McNeill. 1992. Hand and Mind: What Gestures Reveal about Thought. (1992).
- [50] David McNeill. 2005. *Gesture and Thought*. University of Chicago Press, Chicago. doi:10.7208/9780226514642
- [51] Irene Mittelberg. 2008. Peircean semiotics meets conceptual metaphor: Iconic modes in gestural representations of grammar. In *Metaphor and Gesture*, Alan Cienki and Cornelia Müller (Eds.). John Benjamins, Amsterdam, 115–154. doi:10.1075/gs.3.08mit
- [52] Masahiro Mori. 1970. Bukimi no Tani [The Uncanny Valley]. *Energy* 7, 4 (1970), 33–35. Translated by Karl F. MacDorman and Norri Kageki in *IEEE Robotics & Automation Magazine*, 19(2):98–100, 2012.
- [53] OpenAI. 2025. ChatGPT. <https://www.openai.com/chatgpt/>. Accessed 2025-09-02.
- [54] Allan Paivio. 2013. *Imagery and Verbal Processes*. Psychology Press, New York. doi:10.4324/9781315798868
- [55] Lluís Payrató. 2014. Emblems or quotable gestures: Structures, categories, and functions. In *Body – Language – Communication: An International Handbook on Multimodality in Human Interaction*, Vol. 2, Cornelia Müller, Alan Cienki, Ellen Fricke, Silva Ladewig, David McNeill, and Jana Bressem (Eds.). De Gruyter Mouton, Berlin, 1474–1481. doi:10.1515/9783110302028.1474
- [56] Thanawit Prasongpongchai, Pat Pataranutaporn, Monchai Lertsutthiwong, and Pattie Maes. 2025. Talk to the Hand: an LLM-powered Chatbot with Visual Pointer as Proactive Companion for On-Screen Tasks. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 637, 16 pages. doi:10.1145/3706598.3715579
- [57] Matthias Rehm, Yukiko Nakano, Elisabeth André, Toyooki Nishida, Nikolaus Bee, Birgit Endrass, Michael Wissner, Afia Akhter Lipi, and Hung-Hsuan Huang. 2009. From observation to simulation: generating culture-specific behavior for interactive systems. *AI Soc.* 24, 3 (Aug. 2009), 267–280. doi:10.1007/s00146-009-0216-3
- [58] Katharina Reinecke and Abraham Bernstein. 2011. Improving performance, perceived usability, and aesthetics with culturally adaptive user interfaces. *ACM Trans. Comput.-Hum. Interact.* 18, 2, Article 8 (July 2011), 29 pages. doi:10.1145/1970378.1970382
- [59] Qiaoqiao Ren and Tony Belpaeme. 2025. Touch Speaks, Sound Feels: A Multimodal Approach to Affective and Social Touch from Robots to Humans. arXiv:2508.07839 [cs.RO] <https://arxiv.org/abs/2508.07839>
- [60] Maha Salem, Stefan Kopp, Ipke Wachsmuth, Katharina Rohlfing, and Frank Joublin. 2012. Generation and Evaluation of Communicative Robot Gesture. *International Journal of Social Robotics* 4, 2 (2012), 201–217. doi:10.1007/s12369-011-0124-9
- [61] Allison Sauppé and Bilge Mutlu. 2014. Robot Deictics: How Gesture and Context Shape Referential Communication. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction (HRI '14)*. ACM, New York, NY, USA, 342–349. doi:10.1145/2559636.2559657
- [62] Wolfgang Schnotz. 2014. Integrated Model of Text and Picture Comprehension. In *The Cambridge Handbook of Multimedia Learning* (2 ed.), Richard E. Mayer (Ed.). Cambridge University Press, Cambridge, 72–103. doi:10.1017/CBO9781139547369.006
- [63] Hasti Seifi, Arpit Bhatia, and Kasper Hornbæk. 2024. Charting User Experience in Physical Human–Robot Interaction. *J. Hum.-Robot Interact.* 13, 2, Article 27 (June 2024), 29 pages. doi:10.1145/3659058
- [64] Tania Singer, Ben Seymour, John O’Doherty, Holger Kaube, Raymond J Dolan, and Chris D Frith. 2004. Empathy for pain involves the affective but not sensory components of pain. *Science* 303, 5661 (2004), 1157–1162.
- [65] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. 2022. Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (New Orleans, LA, USA) (CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 553, 33 pages. doi:10.1145/3491102.3517719
- [66] Chek Tien Tan, Indriyati Atmosukarto, Budianto Tandianus, Songjia Shen, and Steven Wong. 2025. Exploring the Impact of Avatar Representations in AI Chatbot Tutors on Learning Experiences. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1072, 12 pages. doi:10.1145/3706598.3713456
- [67] Michael Tomasello, Malinda Carpenter, and Ulf Liszkowski. 2007. A new look at infant pointing. *Child Development* 78, 3 (2007), 705–722. doi:10.1111/j.1467-8624.2007.01025.x
- [68] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. LLMR: Real-time Prompting of Interactive Worlds using Large Language Models. arXiv:2309.12276 [cs.HC] <https://arxiv.org/abs/2309.12276>
- [69] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. GesturAR: An Authoring System for Creating Freehand Interactive Augmented Reality Applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology (Virtual Event, USA) (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 552–567. doi:10.1145/3472749.3474769
- [70] Erik Wolf, Sara Klüber, Chris Zimmerer, Jean-Luc Lugin, and Marc Erich Latoschik. 2019. “Paint that object yellow”: Multimodal interaction to enhance creativity during design tasks in VR. In *Proceedings of the 21st ACM International Conference on Multimodal Interaction (ICMI '19)*. ACM, 195–204. doi:10.1145/3340555.3353724

- [71] Pieter Wolfert, Nicole Robinson, and Tony Belpaeme. 2022. A Review of Evaluation Practices of Gesture Generation in Embodied Conversational Agents. *IEEE Transactions on Human-Machine Systems* 52, 3 (2022), 379–389. doi:10.1109/THMS.2022.3149173
- [72] Aaron Wong, Nadine Marcus, Paul Ayres, Lee Smith, Graham A. Cooper, Fred Paas, and John Sweller. 2009. Instructional animations can be superior to statics when learning human motor skills. *Computers in Human Behavior* 25, 2 (2009), 339–347. doi:10.1016/j.chb.2008.12.012
- [73] Zeyi Zhang, Tenglong Ao, Yuyao Zhang, Qingzhe Gao, Chuan Lin, Baoquan Chen, and Libin Liu. 2024. Semantic Gesticulator: Semantics-Aware Co-Speech Gesture Synthesis. *ACM Trans. Graph.* 43, 4, Article 136 (July 2024), 17 pages. doi:10.1145/3658134
- [74] Ada Yi Zhao, Aditya Gunturu, Ellen Yi-Luen Do, and Ryo Suzuki. 2025. Guided Reality: Generating Visually-Enriched AR Task Guidance with LLMs and Vision Models. *arXiv preprint arXiv:2508.03547* (2025). doi:10.48550/arXiv.2508.03547 To appear at UIST 2025.
- [75] Chenfei Zhu, Shao-Kang Hsia, Xiyun Hu, Ziyi Liu, Jingyu Shi, and Karthik Ramani. 2025. agentAR: Creating Augmented Reality Applications with Tool-Augmented LLM-based Autonomous Agents. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 54, 23 pages. doi:10.1145/3746059.3747676

## A Object Registration

The Object Registration module (Sec 4.2) recognizes physical objects from an RGB image and, using auxiliary spatial data, infers each object's dimensions and orientation. As described in Sec 4.2, we partition each object into six faces and nine regions per face to locate potential functional or semantic areas (e.g., a filament mounted on a 3D printer, a blossom on a plant). The Object Registration System Prompt instructs the agent to emit a structured object description that conforms to our predefined schema, enabling automatic registration in the object registry and extraction of key fields to append to the Response Generation System Prompt. Below, we detail the four key sub-modules to complete the entire object registration process.

### A.1 Auxiliary Spatial Data Pre-Processing (local XR headset)

When a user prompts to register this object, we have the real-time eye gaze ray intersecting with the scene mesh to get a 3D position. Meanwhile, using the camera's intrinsics and extrinsics, we project this intersection point onto the captured RGB image coordinate space as the Projected Intersection Point (via the turn channel), which will be sent together with RGB Image (via the image channel) to the LLM engine.

### A.2 Object Registration System Prompt (remote LLM engine)

**\*\*Role:\*\*** You are an expert visual perception and object semantics agent. Your primary function is to analyze first-person visual data to identify an object and provide its canonical 2D and 3D semantic definitions.

**\*\*Context (Input):\*\*** You will receive a data packet containing:

- (1) RGB Image: The user's current view.
- (2) Projected Intersection Point: The precise 2D coordinate  $[x, y]$  on the image that the user is focusing on. Use this as a hint to identify the primary object of interest.

**\*\*Core Task:\*\*** Based on the provided context, you must analyze the object at the Projected Intersection Point and perform the following four critical tasks. Your entire response MUST be formatted as a single, valid JSON object.

- (1) Object Identification (objectName): Analyze the visual data. Provide a specific, concrete, camelCase identifier for the object (e.g., orchidPlant, prusaPrinter, redBullCan, officeDesk).
- (2) 2D Bounding Box (boundingBox2D): Based on the RGB Image and the Projected Intersection Point hint, generate a 2D bounding box  $[xmin, ymin, xmax, ymax]$  that tightly encloses the identified object in pixel coordinates.
- (3) Dimension Estimation (dimensions): Based ONLY on your general semantic knowledge of the identified object, estimate its canonical, real-world average 3D dimensions in meters. Use a width (X-axis), height (Y-axis), and depth (Z-axis, thickness) format. This is crucial for the local system to calculate the object's 3D volumetric center.
- (4) Semantic Definition: This task defines the object's functional parts relative to its orientation.
  - (4-1) Orientation Assumption: You MUST generate the semanticMap by assuming the user's perspective in the RGB Image is the object's canonical "Front" (F). All other faces (B-Back, T-Top, D-Down, L-Left, R-Right) are defined relative to this assigned "Front".
  - (4-2) Object Local Region Format: All local regions MUST use the encoded string format:  $\{face\}_{vRegion}\{hRegion\}$ .
    - $\{face\}$ : One of F (Front), B (Back), T (Top), D (Down), L (Left), R (Right).
    - $\{vRegion\}$ : Vertical region on the face, one of top, middle, bottom.
    - $\{hRegion\}$ : Horizontal region on the face, one of front, center, right.
    - Examples: F\_middle\_center, T\_top\_right, L\_bottom\_left.
  - (4-3) Default Region (defaultRegion): Specify the single, most common region to point at when referring to the object in general, using the format above (e.g., F\_middle\_center).
  - (4-4) Semantic Map (semanticMap): Generate a JSON object. Each key must be the camelCase name of a functional part. The value for each key must be a JSON object containing:
    - location: The string-encoded location (e.g., F\_middle\_center).
    - description: A brief, user-facing description of the part's function (e.g., "The flat bed where the object is printed.").

**\*\*Strict Output Format (JSON):\*\*** Your output must be ONLY the JSON object, with no other text before or after it.

### Example JSON Output Format:

```
{
  "objectName": "exampleObject",
  "boundingBox2D": [0, 0, 0, 0],
  "dimensions": {
    "width": 0.0,
    "height": 0.0,
    "depth": 0.0
  },
  "defaultRegion": "F_middle_center",
  "semanticMap": {
    "partNameOne": {
      "location": "F_vRegion_hRegion",
      "description": "A brief description of this part."
    },
    "partNameTwo": {
      "location": "T_vRegion_hRegion",
      "description": "A brief description of this part."
    }
  }
}
```

### Example: 3D Printer

```
{
  "objectName": "3dPrinter",
  "boundingBox2D": [120, 300, 800, 900],
  "dimensions": {
    "width": 0.4,
    "height": 0.5,
    "depth": 0.45
  },
  "defaultRegion": "F_middle_center",
  "semanticMap": {
    "printBed": {
      "location": "F_middle_center",
      "description": "The flat, heated square bed where the object is printed."
    },
    "nozzle": {
      "location": "F_top_center",
      "description": "The hot metal tip that extrudes the melted filament."
    },
    "controlScreen": {
      "location": "F_bottom_left",
      "description": "The LCD interface used to control the printer."
    },
    "cardReader": {
      "location": "F_bottom_right",
      "description": "The SD card slot for loading print files."
    },
    "filamentSpool": {
      "location": "R_top_center",
      "description": "The spool that holds the plastic filament."
    }
  }
}
```

```
},
  "filamentIntake": {
    "location": "F_top_center",
    "description": "The port where the filament enters the extruder."
  }
}
}
```

### Example: Orchid Plant

```
{
  "objectName": "orchidPlant",
  "boundingBox2D": [340, 210, 680, 750],
  "dimensions": {
    "width": 0.15,
    "height": 0.4,
    "depth": 0.15
  },
  "defaultRegion": "T_middle_center",
  "semanticMap": {
    "flower": {
      "location": "T_middle_center",
      "description": "The decorative blooms of the plant."
    },
    "stem": {
      "location": "F_middle_center",
      "description": "The main stalk supporting the flowers."
    },
    "rootArea": {
      "location": "F_bottom_center",
      "description": "The base of the plant, including visible roots."
    },
    "potRim": {
      "location": "T_bottom_center",
      "description": "The top edge of the pot containing the plant."
    }
  }
}
```

### Example: Office Whiteboard

```
{
  "objectName": "officeWhiteboard",
  "boundingBox2D": [50, 150, 910, 800],
  "dimensions": {
    "width": 1.2,
    "height": 0.9,
    "depth": 0.05
  },
  "defaultRegion": "F_middle_center",
  "semanticMap": {
    "writingSurface": {
      "location": "F_middle_center",
      "description": "The main, glossy white area for writing."
    },
    "markerTray": {
      "location": "F_bottom_center",
    }
  }
}
```

```

    "description": "The horizontal ledge at the bottom
    that holds markers."
  },
  "frameEdge": {
    "location": "L_middle_center",
    "description": "The metal or plastic border
    surrounding the surface."
  }
}
}
}

```

### A.3 Response Processing (local XR headset)

We need to process the agent output to generate the oriented and scaled bounding box of the object. Given the 2D bounding box coordinates, the estimated object dimensions, the head position, and the original 3D intersection point, we are able to anchor the object's bounding box in the XR space. In the algorithm shown below, the `ScreenPointToRay()` and `RayPlaneIntersection()` are supported by commercial game engines such as Unity3D.

---

#### Algorithm 1 3D Bounding Box Generation

---

**Require:** Object registration agent response:  
 $bbox_{2D}$ : The image coordinates of the estimated 2D bounding box (i.e.,  $[x_{min}, y_{min}, x_{max}, y_{max}]$ ).  
 $dims$ : The estimated 3D dimensions of the object (i.e.,  $width, height, and depth$ ).

**Require:** Local spatial data:  
 $P_{head}$ : Head Position (World Space)  
 $M$ : Camera View Matrix  
 $K$ : Camera Intrinsic Matrix  
 $P_{intersect\_3D}$ : 3D Intersection Point (World Space)  
 $P_{intersect\_2D}$ : 2D Intersection Point (Screen Space)  
 $V_{world\_up}$ : Global Up Vector (e.g.,  $[0, 1, 0]$ )

**Ensure:** 3D bounding box transform  
 $P$ : Center Position of the Bounding Box  
 $R$ : Orientation Matrix of the Bounding Box  
 $S$ : Size of the Bounding Box

- 1: **procedure** GETBOUNDINGBOX
- 2:  $V_{gaze} \leftarrow P_{head} - P_{intersect\_3D}$
- 3:  $V_{fwd\_raw} \leftarrow [V_{gaze}.x, 0, V_{gaze}.z]$
- 4:  $V_{forward} \leftarrow normalize(V_{fwd\_raw})$
- 5:  $V_{up} \leftarrow V_{world\_up}$
- 6:  $V_{right} \leftarrow normalize(cross\_product(V_{up}, V_{forward}))$
- 7:  $R \leftarrow [V_{right}, V_{up}, V_{forward}]$
- 8:
- 9:  $P_{2D\_center} \leftarrow [(bbox_{2D}.x_{min} + bbox_{2D}.x_{max})/2, (bbox_{2D}.y_{min} + bbox_{2D}.y_{max})/2]$
- 10:  $Ray_{cam} \leftarrow ScreenPointToRay(P_{2D\_center}, P_{head}, M, K)$
- 11:  $\Pi_{front} \leftarrow define\_plane(P_{intersect\_3D}, V_{forward})$
- 12:  $P_{center\_front} \leftarrow RayPlaneIntersection(Ray_{cam}, \Pi_{front})$
- 13:  $d \leftarrow dims.depth/2$
- 14:  $P \leftarrow P_{center\_front} - (V_{forward} \cdot d)$
- 15:
- 16:  $S \leftarrow [dims.width, dims.height, dims.depth]$
- 17:
- 18: **return**  $P, R, S$
- 19: **end procedure**

---

### A.4 Object Registry (local XR headset)

As addressed in Sec 4.2, each processed object with the label, transform, and semantic maps will be added to the object registry for the `GestureEvents` to read.

Separately, to inform the main agent with all available objects, we have a custom function to generate a string of available objects (i.e., only fetch the `objectName` and `semanticMap` to build a string) and append it to the Response Generation System Prompt introduced below. Here is an example generated Registered Objects string. Note that this string will be updated on the runtime.

```

## Registered Objects
The following objects and their named semantic parts are
currently registered in the scene. You MUST treat
this as the complete and exclusive list of objects
and functional regions you can refer to.

**Object:** `3dPrinter`

```

- `printBed` (`F\_middle\_center`): The flat, heated square bed where the object is printed.
- `nozzle` (`F\_top\_center`): The hot metal tip that extrudes the melted filament.
- `controlScreen` (`F\_bottom\_left`): The LCD interface used to control the printer.
- `cardReader` (`F\_bottom\_right`): The SD card slot for loading print files.
- `filamentSpool` (`R\_top\_center`): The spool that holds the plastic filament.
- `filamentIntake` (`F\_top\_center`): The port where the filament enters the extruder.

```

**Object:** `orchidPlant`
- `flower` (`T_middle_center`): The decorative blooms of the plant.
- `stem` (`F_middle_center`): The main stalk supporting the flowers.
- `rootArea` (`F_bottom_center`): The base of the plant, including visible roots.
- `potRim` (`T_bottom_center`): The top edge of the pot containing the plant.

```

## B GestureEvents Definition and Few-Shot Examples

Here is the prompt for:

- (1) Defining the GestureEvent with the component breakdown.
- (2) Available attributes and corresponding values.
- (3) All supported GestureEvents.
- (4) Few-shot examples.

Note that when defining the attribute for physical objects, we instruct the agent to refer to the Registered Scene Objects (see the above section).

```

## Gesture Definitions & Rules

**Syntax Definition:**
You must embed gestures using the following strict format:

`[id; attribute1:value1, attribute2:value2, ...]keyword`

**Component Breakdown:**
1. **`[id; attribute1:value1, attribute2:value2, ...]`**:
   The gesture event definition is enclosed in square brackets.
2. **`id`**: This is the mandatory, unique string identifier for the gesture.
3. **`;`**: A mandatory semicolon *must* follow the `id`.
4. **`attribute1:value1, attribute2:value2, ...`**: A comma-separated list of required parameters.
5. **Token-Saving Rule:** You MUST omit any attribute that uses one of the Global Default Values.
6. **`keyword`**: The single, specific word in your response, immediately following the `]`, that acts as the trigger.

---

**Master Parameter List & Global Defaults:**
* `handedness`: **single** (Valid: single, double)

```

```

* `gesture`: **palm** (Valid: palm, fist, point, hold, grab, thumbUp, thumbDown, one, two, three, cylinder, cube, sphere)
* `dynamics`: **static** (Valid: static, rollRotation, pitchRotation, yawRotation, xTranslation, yTranslation, zTranslation)
* `Interactivity`: **none** (Valid: none, highFive, shakeHand, holdHand)
* `Vfx`: **none** (Valid: none, burn, freeze, glow, poisoned, spoil, explode)
* `spatiality`: **midAir**
  * **RULE:** This is the default. If a gesture targets an object, the `spatiality` value **must** be a specific object coordinate in the format `[objectName]_[face]_[vRegion]_[hRegion]`. You **MUST** derive this coordinate by cross-referencing the `Registered Objects` list provided.

```

---

```

**Supported GestureEvents List (Must Use Only These IDs):**

```

```

**A. Deictic Gestures (Pointing/Showing)**
* **1. PointObject:** Static point at a whole object.
  * Requires: `gesture:point`.
  * Requires: `spatiality`. The value MUST be the object's **Default Region** from the `Registered Scene Objects` list (e.g., `spatiality:3dPrinter_F_middle_center`).
* **2. RegionOfObject:** Static point at a *specific semantic region* of an object.
  * Requires: `gesture:point`.
  * Requires: `spatiality`. You MUST find the mentioned part (e.g., "card reader") in the `Registered Scene Objects` list and use its precise mapped coordinate (e.g., `spatiality:3dPrinter_F_bottom_right`).
* **3. DirectionOfObject:** Dynamic point to draw focus to an object.
  * Requires: `gesture:point`, `dynamics:yTranslation`.
  * Requires: `spatiality` (Use the object's *Default Region* from the list, e.g., `spatiality:component_A_F_middle_center`).

**B. Iconic Gestures (Representing Actions/Forms)**
* **4. ObjectProxy:** Mimics a 3D shape.
  * Requires: `gesture:[cube|sphere|cylinder]`, `dynamics:yawRotation`.
* **5. Action:** Mimics a physical manipulation.
  * Requires: `gesture:[hold|grab]`, `dynamics:[pitchRotation|zTranslation|etc.]`.
  * Optional: `spatiality:...` (if the action targets a specific, registered object part).
* **6. Geometry1D:** Represents a 1D line/length.
  * Requires: `gesture:point`, `dynamics:xTranslation`.
* **7. Geometry2D:** Represents a 2D area.

```

```

* Requires: `handedness:double`, `dynamics:xTranslation`
`.

* **8. Geometry3D:** Represents a 3D volume.
* Requires: `handedness:double`, `gesture:sphere`.

* **9. Numeric:** Shows a number count.
* Requires: `gesture:[one|two|three]`.

**C. Expression Gestures (Emotive/Interactive)**
* **10. Positive:** Approval.
* Requires: `gesture:thumbUp`. Optional: `handedness:double`.

* **11. Negative:** Disapproval.
* Requires: `gesture:thumbDown`. Optional: `handedness:double`.

* **12. Attention:** General "look here" gesture in mid-air.
* Requires: `gesture:point`, `dynamics:yTranslation`. (Defaults to `spatiality:midAir`).

* **13. Prohibition:** "Don't do that" (finger wag).
* Requires: `gesture:point`, `dynamics:yawRotation`.

* **14. Comparison:** Show difference between two things.
* Requires: `handedness:double`, `dynamics:zTranslation`. Optional: `spatiality:...` (to compare against a registered object).

* **15. Celebration:** Interactive high-five.
* Requires: `Interactivity:highFive`.

* **16. Burn / Poisoned / Spoil:** Apply a visual effect.
* Requires: `Vfx:[burn|poisoned|spoil]`. Optional: `spatiality:...` (to apply to a registered object region).

```

## B.1 Few-Shot Examples

```

## Few-Shot Examples (Adhering to the Context Above)

**User:** How do I use this 3D printer?
**Assistant:** This [PointObject; gesture:point, spatiality:3dPrinter_F_middle_center]printer is simple to use. First, check the [RegionOfObject; gesture:point, spatiality:3dPrinter_R_top_center]filament spool. Then, feed the end of the filament into this [RegionOfObject; gesture:point, spatiality:3dPrinter_F_top_center]intake. Finally, put your SD card in the [RegionOfObject; gesture:point, spatiality:3dPrinter_F_bottom_right]card reader here.

**User:** How do I care for this orchid?

```

```

**Assistant:** Orchids need special care. You only need to [Action; gesture:hold, dynamics:pitchRotation, spatiality:orchidPlant_F_bottom_center]pour a small amount of water directly onto the [RegionOfObject; gesture:point, spatiality:orchidPlant_F_bottom_center]root area, which is at the base. Be careful, [Prohibition; gesture:point, dynamics:yawRotation]don't let water sit on the [RegionOfObject; gesture:point, spatiality:orchidPlant_T_middle_center]flowers.

**User:** Is it safe to touch this component?
**Assistant:** [Attention; gesture:point, dynamics:yTranslation]Whoa, absolutely [Negative; gesture:thumbDown]not. Do not touch [DirectionOfObject; gesture:point, spatiality:component_A_F_middle_center, dynamics:yTranslation]that part when the light is red. It will [Burn; Vfx:burn, spatiality:component_A_F_middle_center]burn you.

**User:** What's the difference between these two boxes?
**Assistant:** The [PointObject; gesture:point, spatiality:box_A_T_middle_center]first box has a wide [Geometry2D; handedness:double, dynamics:xTranslation]surface on top. The [PointObject; gesture:point, spatiality:box_B_F_middle_center]second box is much narrower but has this [RegionOfObject; gesture:point, spatiality:box_B_T_top_right]antenna. The main difference is that the [Comparison; handedness:double, spatiality:box_A_T_middle_center, dynamics:zTranslation]second box must be positioned further from the wall than the first.

```

## C Response Generation System Prompt

After we introduced the object registration, and the `GestureEvent` definition, here is the central system prompt to complete the response augmentation task. The two above-mentioned key sub-components that will be appended below this main prompt.

```

**Role:** You are an embodied assistive AI. Your primary function is to provide helpful, descriptive text responses to user queries about spatial tasks, 3D objects, and procedures.

**Core Directive:** Your textual responses MUST be augmented with embedded `GestureEvent` commands. These commands are parsed by a robotic hand agent system to synchronize physical gestures with your spoken words.

**Context and Rules:** You must operate according to the following two blocks of information which will be appended directly after this instruction:
1. **Registered Objects:** A dynamic list of all objects currently recognized in the scene and their named parts. This is your absolute ground truth for all spatial targeting and the `spatiality` attribute.

```

2. **\*\*Gesture Definitions & Examples:\*\*** Your complete static instruction set. This includes the mandatory syntax, all allowed gesture commands, parameter definitions, and few-shot examples that you must follow.

Your task is to generate a response that strictly adheres to all rules and context provided in those blocks.

## D Agent Transcripts of the User Study Tasks

This prototype is a turn-based conversational system, fully decoupled from the backend server. We use local STT only to detect that the user has spoken and then advance through a scripted transcript. Below, we present both user and assistant turns; the user utterances are illustrative placeholders rather than free input. During the tasks, spontaneous user speech was not processed. At the start of the study, participants were instructed to respond only with the phrases prompted by the assistant. Assistant prompts were written to include gentle hand-offs (e.g., “Would you like to know more?” “Shall we continue?”) to cue participants to trigger the next step when ready.

As we mentioned in Sec 6, each task contains 10 responses where each response may contain more than one `GestureEvents`. There are 37 `GestureEvents` in the Orchid Care task, and 29 `GestureEvents` in the 3D Printer Operation task.

### D.1 Task 1: Orchid Care

**User:** “I just got this new plant, tell me something about it.”

**Assistant:** “That’s a [`PointObject`; `gesture:point`, `spatiality:orchidPlant_T_middle_center`]beautiful orchid! Let’s take a closer look at it. Let me introduce you to the orchid by its parts. Let me know when you are ready.”

**User:** “Sure.” (Or express intention to continue the conversation)

**Assistant:** “Look at [`RegionOfObject`; `gesture:point`, `spatiality:orchidPlant_T_middle_center`]those amazing flowers. That’s definitely the most eye-catching part! And they can last for around five weeks! See [`RegionOfObject`; `gesture:point`, `spatiality:orchidPlant_F_middle_center`]that stem? That’s what we call the flower spike, and it’s what holds up all those gorgeous blooms. You want to know more?”

**User:** “Sure.” (Or express intention to continue the conversation)

**Assistant:** “Now, check out [`RegionOfObject`; `gesture:point`, `spatiality:orchidPlant_F_bottom_center`]the leaves. Notice how [`Geometry2D`; `handedness:double`, `dynamics:xTranslation`]thick and kind of fleshy they are? Those leaves are actually designed to [`Geometry3D`; `handedness:double`, `gesture:sphere`]store water for the plant, which is pretty clever. Those leaves look healthy and vibrant, which is a [`Positive`; `gesture:thumbUp`]happy sign. You see some [`RegionOfObject`; `gesture:point`, `spatiality:orchidPlant_F_bottom_center`]roots peeking out of the pot? Those are called air roots, and actually help the plant [`Action`; `gesture:grab`]grab moisture and nutrients right out of the air. Would you like to learn about how to care for your orchid?”

**User:** “Yes, please tell me how to care for it.” (Or express intention to continue the conversation)

**Assistant:** “Okay let’s learn this step by step. Step [`Numeric`; `gesture:one`]one is about [`Action`; `gesture:hold`, `dynamics:pitchRotation`]watering the orchid. [`Attention`; `gesture:point`, `dynamics:yTranslation`]Now let’s start our first watering! Are you ready?”

**User:** “Yes, I’m ready.” (Or express intention to continue the conversation)

**Assistant:** “First, take your [`PointObject`; `gesture:point`, `spatiality:waterBottle_F_middle_center`]watering bottle, and slowly pour room-temperature [`Action`; `gesture:hold`, `dynamics:pitchRotation`, `spatiality:orchidPlant_F_bottom_center`]water through the potting mix. Keep pouring until you see water starting to drain out of the [`RegionOfObject`; `gesture:point`, `spatiality:orchidPlant_F_bottom_center`]bottom of the pot.”

**User:** “Got it!” (Or express intention to continue the conversation)

**Assistant:** “[`Attention`; `gesture:point`, `dynamics:yTranslation`]The most important part here is to [`Action`; `gesture:grab`, `spatiality:orchidPlant_T_bottom_center`, `dynamics:yTranslation`]lift it and let it drain completely. You absolutely [`Prohibition`; `gesture:point`, `dynamics:yawRotation`]do not want [`ObjectProxy`; `gesture:cylinder`, `dynamics:yawRotation`]your orchid sitting in standing water to get [`Negative`; `gesture:thumbDown`]root rot. So, [`Attention`; `gesture:point`, `dynamics:yTranslation`]just let all that excess water run out.”

**User:** “Okay.” (Or express intention to continue the conversation)

**Assistant:** “[`Attention`; `gesture:point`, `dynamics:yTranslation`]We also need to fertilize the Orchid. Let’s look at the [`PointObject`; `gesture:point`, `spatiality:orchidFood_F_middle_center`]fertilizer spray on the table. We can apply the fertilizer about [`Numeric`; `gesture:one`]once a month when your orchid is actively growing. Are you ready to apply some orchid food?”

**User:** “Okay.” (Or express intention to continue the conversation)

**Assistant:** “Great! Let’s get started. First, take the [`PointObject`; `gesture:point`, `spatiality:orchidFood_F_middle_center`]fertilizer spray and give it a [`Action`; `gesture:hold`, `spatiality:orchidFood_F_middle_center`, `dynamics:rollRotation`]good shake. Then, [`Action`; `gesture:point`, `dynamics:zTranslation`]spray a light mist onto the [`RegionOfObject`; `gesture:point`, `spatiality:orchidPlant_F_bottom_center`]leaves and any exposed roots. About four to five shots in total. [`Prohibition`; `gesture:point`, `dynamics:yawRotation`]Don’t spray too much, it will [`Burn`; `Vfx:burn`]burn the orchid.”

**User:** “Got it!” (Or express intention to continue the conversation)

**Assistant:** “Alright, this is the fun part. Let’s find the absolute best spot for your new orchid in the room! Given that [`PointObject`; `gesture:point`, `spatiality:orchidPlant_T_middle_center`]the orchid [`Positive`; `gesture:thumbUp`]loves bright, indirect light, how about we try this: Let’s find a good spot near [`DirectionOfObject`; `gesture:point`, `dynamics:yTranslation`, `spatiality>window_F_middle_center`]that window. Let’s place the Orchid about [`Geometry1D`; `gesture:point`, `dynamics:xTranslation`]thirty centimeters away from

the window. That’s usually [Positive; gesture:thumbUp]perfect because it gets that lovely, gentle morning sun.”

**User:** “Got it.” (Or express intention to continue the conversation)

**Assistant:** “[Positive; gesture:thumbUp]Good Job! Let’s [Celebration; Interactivity:highFive]Hi Five.”

## D.2 Task 2: 3D Printer Operation

**User:** “Can you introduce this 3D printer to me?”

**Assistant:** “I see that you have an [PointObject; gesture:point, spatiality:3dPrinter\_F\_middle\_center]Prusa Original MK3 which is a [Positive; gesture:thumbUp]fantastic printer. Before we even get to printing, [Attention; gesture:point, dynamics:yTranslation]let’s just take a quick tour around your [PointObject; gesture:point, spatiality:3dPrinter\_F\_middle\_center]Prusa printer so you know what everything is. Are you ready?”

**User:** “Sure.” (Or express intention to continue the conversation)

**Assistant:** “First, see that [RegionOfObject; gesture:point, spatiality:3dPrinter\_F\_middle\_center]plate at the bottom? That’s your Print Bed. This is where your actual 3D object will [ObjectProxy; gesture:cylinder, dynamics:yawRotation]be built, layer by layer.”

**User:** “Sure.” (Or express intention to continue the conversation)

**Assistant:** “Now, look at the [RegionOfObject; gesture:point, spatiality:3dPrinter\_F\_top\_center]print head above the bed. At the very tip of it, that little pointy metal part, is the Nozzle. This is where the melted plastic [Action; gesture:point, dynamics:zTranslation]comes out.”

**User:** “Okay.” (Or express intention to continue the conversation)

**Assistant:** “Next, you’ll see a [RegionOfObject; gesture:point, spatiality:3dPrinter\_R\_top\_center]roll of plastic filament. This is your Filament Spool. This is the raw material, the plastic ink for your 3D prints!”

**User:** “Okay.” (Or express intention to continue the conversation)

**Assistant:** “Finally, on the [RegionOfObject; gesture:point, spatiality:3dPrinter\_F\_bottom\_left]front of the printer, you’ll see a small screen and a rotating knob. This is your Control Panel. [Action; gesture:hold, spatiality:3dPrinter\_F\_bottom\_left, dynamics:rollRotation]This is how you’ll communicate with the printer. Are you ready to print something?”

**User:** “Yes.” (Or express intention to continue the conversation)

**Assistant:** “First, [Action; gesture:point, dynamics:zTranslation]turn on your printer if it’s not already on. You’ll usually find the power switch [RegionOfObject; gesture:point, spatiality:3dPrinter\_B\_bottom\_right]near the back side.”

**User:** “Okay.” (Or express intention to continue the conversation)

**Assistant:** “Now, grab that [PointObject; gesture:point, spatiality:sdCard\_F\_middle\_center]SD card from the desk. You’ll see an SD card slot on [RegionOfObject; gesture:point, spatiality:3dPrinter\_F\_bottom\_right]the side of the control panel. Carefully [Action; gesture:grab, spatiality:3dPrinter\_F\_bottom\_right, dynamics:zTranslation]insert the SD card into the slot. Make sure it goes in the right way.”

**User:** “Okay.” (Or express intention to continue the conversation)

**Assistant:** “[Positive; gesture:thumbUp]Great! Now the screen should show the files in that SD card. You’ll use the knob to [Action; gesture:hold, spatiality:3dPrinter\_F\_bottom\_left, dynamics:rollRotation]scroll through the menu options and [Action; gesture:point, spatiality:3dPrinter\_F\_bottom\_left, dynamics:zTranslation]press it in to select the file.”

**User:** “Got it!” (Or expresses intention to continue the conversation)

**Assistant:** “Alright, [Attention; gesture:point, dynamics:yTranslation]this is the fun part. The printer will then begin its heating sequence. It will heat up both the [RegionOfObject; gesture:point, spatiality:3dPrinter\_F\_middle\_center]print bed and [RegionOfObject; gesture:point, spatiality:3dPrinter\_F\_top\_center]the nozzle to the required temperatures for your filament. You’ll see those temperatures rising on [RegionOfObject; gesture:point, spatiality:3dPrinter\_F\_bottom\_left]the screen. In around [Numeric; gesture:one]one minute the temperatures will be reached, the printer will [Positive; gesture:thumbUp]start! [Prohibition; gesture:point, dynamics:yawRotation]Don’t touch the print bed or nozzle, they are [Burn; Vfx:burn]hot!”

**User:** “Got it.” (Or express intention to continue the conversation)

**Assistant:** “[Positive; gesture:thumbUp]Good Job! Let’s [Celebration; Interactivity:highFive]Hi Five.”

## E Participant Information

**Table 1: Participant demographics for the formative study (N=10)**

ID	Expertise	Age	XR	LLM
P1	UX Researcher	28	Used	Freq.
P2	XR Developer	31	Built	Freq.
P3	XR Developer	35	Built	Freq.
P4	XR Developer	29	Built	Built
P5	UX Researcher	34	Used	Freq.
P6	XR Developer	32	Built	Freq.
P7	XR Developer	38	Built	Built.
P8	UX Researcher	36	Used	Freq.
P9	XR Developer	30	Built	Built.
P10	UX Researcher	37	Used	Freq.

Gender: 7 Male, 3 Female

Ethnicity: 3 East Asian, 3 White, 2 Black, 2 South Asian

Age: M = 33.0, SD = 3.49, Range = 28–38

XR: Built = Developed XR apps, Used = Used XR apps

LLM: Built = Developed LLM apps, Freq. = Frequent use,

Occ. = Occasional use

**Table 2: Participant demographics for the user study (N=12)**

ID	Age	Occupation	XR	LLM	Spatial Task
U1	37	SWE	Used	Occ.	Y
U2	19	SWE	Built	Freq.	N
U3	30	UXR	Used	Freq.	Y
U4	28	SWE	Built	Freq.	Y
U5	30	HCI	Built	Built	N
U6	24	UXD	Used	Freq.	N
U7	37	SWE	Built	Freq.	N
U8	26	UXR	Never	Freq.	N
U9	36	HCI	Used	Freq.	N
U10	30	UXD	Built	Freq.	Y
U11	30	SWE	Used	Freq.	N
U12	29	UXR	Used	Built	Y

Gender: 10 Male, 1 Female, 1 Non-binary

Ethnicity: 4 East Asian, 4 White, 2 South Asian, 2 Black

Age: M = 29.7, SD = 5.3, Range = 19–37

Occupation: SWE = Software Engineer, UXD = UX Designer,

UXR = UX Researcher, HCI = HCI Researcher

XR: Built = Developed XR apps, Used = Used XR apps, Never = No experience

LLM: Built = Developed LLM apps, Freq. = Frequent use, Occ. = Occasional use

Spatial Task = Has used LLM assistants for spatially grounded tasks (Y/N)