VsensAgent: A Human-Agent Interface for Collaborative Virtual Sensor Data Authoring

FENGZHOU LIANG*, Keio University, Japan
TIAN MIN*, Keio University, Japan
CHENGSHUO XIA, Xidian University, China and Keio University, Japan
YUTA SUGIURA†, Keio University, Japan

Virtual sensing can reduce the cost of generating sensor datasets for human activity recognition and ubiquitous sensing, yet users must still translate sensing intent into concrete scene configurations. This fragmentation makes it difficult to translate sensing goals into concrete configurations and refine them as results emerge. We present *VsensAgent*, a human–agent interface that grounds an LLM-powered agent in a shared 3D simulation environment. Users combine natural-language delegation with direct scene manipulation, while the agent can inspect scene state, configure sensors and avatar motions, record data, and support signal analysis. We conducted a task-based study of *VsensAgent* in which 12 participants completed 36 virtual sensing authoring sessions. Analysis of interaction logs, authored configurations, and post-task interviews showed that participants progressively developed intent through executable scene states, shifted control between agent delegation and direct manipulation at the level of individual operations, and treated agent proposals as editable starting points. These findings frame virtual sensor generation as progressive, collaborative authoring and inform the design of agents for stateful workflows.

CCS Concepts: • **Human-centered computing → User interface toolkits**.

Additional Key Words and Phrases: User Interface, Virtual Sensors, AI Agent, Human–Agent Collaboration, Simulation

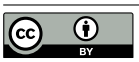
ACM Reference Format:
Fengzhou Liang, Tian Min, Chengshuo Xia, and Yuta Sugiura. 2026. VsensAgent: A Human-Agent Interface for Collaborative Virtual Sensor Data Authoring. *Proc. ACM Hum.-Comput. Interact.* 10, 8, Article ISS025 (November 2026), 27 pages. https://doi.org/10.1145/3859337

1 Introduction

Sensor datasets underpin human activity recognition (HAR) and ubiquitous sensing systems, but collecting them in the physical world remains costly. Researchers must recruit participants, instrument bodies or environments, and subsequently annotate, synchronize, and clean the resulting sensor streams [6, 7]. Virtual sensing offers an alternative by synthesizing sensor signals from simulated environments, avatars, and animations [18, 19, 38]. It enables configurations and activities to be varied without repeating physical data collection and has supported applications ranging

*Both authors contributed equally to the paper.
†Yuta Sugiura is the corresponding author.

Authors' Contact Information: Fengzhou Liang, Lifestyle Computing Lab, Keio University, Yokohama, Japan, liangfz@keio.jp; Tian Min, Lifestyle Computing Laboratory, Keio University, Yokohama, Kanagawa, Japan, welkinmin@keio.jp; Chengshuo Xia, Guangzhou Institute of Technology, Xidian University, Guangzhou, China and Keio University, Yokohama, Japan, csxia@keio.jp; Yuta Sugiura, Keio University, Yokohama, Japan, sugiura@keio.jp.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 2573-0142/2026/11-ARTISS025
https://doi.org/10.1145/3859337

from exercise recognition to accessibility [25, 27, 47]. However, the technical ability to synthesize signals does not make virtual sensor data easy to author.

A typical virtual sensing workflow remains distributed across 3D editors, scripts, and separate visualization tools. Users must translate a sensing goal into sensor modalities, placements, orientations, avatar motions, and analysis procedures. Integrated platforms have reduced some of this tooling fragmentation [28, 29, 46], yet authoring must bridge sensing intent and scene specification: users may know what they want to observe without knowing how to configure the simulation to capture it. Relevant constraints often become apparent only after a configuration has been executed and its spatial or signal-level consequences inspected. We therefore frame virtual sensor generation not simply as parameter specification, but as an iterative authoring process in which sensing intent is progressively made concrete.

Language-model-based agents offer a possible interface for this process by combining conversational reasoning with tool use [3, 30, 52]. In virtual sensing, however, useful assistance requires more than generating instructions or executing a complete request. The sensing configuration may evolve through dialogue, agent actions, manual edits, and analysis feedback. An agent must remain grounded in this shared state, while users need to inspect its actions or directly revise the resulting scene. This raises an interaction question beyond pipeline automation: how can users and agents jointly develop a sensing configuration when intent, control, and intermediate artifacts evolve together?

Informed by formative interviews with four virtual sensing practitioners, we developed *VsensAgent*, an open-sourced human–agent interface [1] for authoring virtual sensor data in a shared 3D simulation environment. Users can describe goals and delegate operations through natural language or directly manipulate sensors and avatars in the scene. The agent observes the evolving scene state and can configure sensors, generate avatar motions, record virtual sensor data, and delegate signal analysis to a subagent. By grounding language and direct manipulation in the same environment, *VsensAgent* supports an iterative movement from sensing intent to executable configuration, and with observed results informing subsequent revisions.

To understand how this form of collaboration unfolds in practice, we conducted a user study with 12 participants, who completed three authoring tasks for a total of 36 sessions. We analyzed interaction logs, authored configurations, screen recordings, and post-task interviews, tracing the authoring process from participants' initial interpretation of the task to their final configurations. Our findings characterize virtual sensor authoring as a progressive collaborative process, in which sensing goals become concrete through interaction with the agent and an executable scene. From this account, we derive implications for designing human–agent interfaces that support evolving intent, shared control, and exploration in stateful authoring workflows. This work makes three contributions:

- *VsensAgent*, a human–agent interface that integrates conversational assistance, direct scene manipulation, virtual sensor recording, and signal analysis within a shared simulation state.
- A remote task-based study of 12 participants across 36 authoring sessions, providing an empirical account of human–agent collaboration in virtual sensor data authoring.
- Design implications concerning progressive specification, operation-level co-agency, and adaptive scaffolding in under-specified design spaces.

[1] Avaliable at: https://github.com/KEIO-LCLAB/VsensAgent-public

2 Related Work

2.1 Embodied Agents and Simulation Platforms

Embodied AI situates an agent within an environment that it can perceive and affect. Such embodiment may be physical or virtual; its defining property is a perception–action loop in which reasoning is conditioned on environment state and actions produce subsequent observations [10, 23]. Simulation platforms such as Gibson [50] or AI2-THOR [17] provide this coupling through structured 3D scenes, controllable objects, and executable skills. More recent language-model-based systems extend virtual agents with memory, planning, and natural-language interaction, for example, placing LLM-driven characters in an interactive sandbox where their observations and actions accumulate into persistent behavioral context [36]. Across these systems, the virtual environment provides action-relevant context that cannot be represented by conversation alone.

Simulation has most commonly served as an arena in which autonomous agents are trained, evaluated, or observed. Recent HCI work has begun to treat interactive environments as artifacts that users and generative systems construct together. For example, LLMR [9] combines scene understanding, task planning, and code generation to create and modify interactive mixed-reality worlds through language; ImaginateAR [21] similarly grounds an AI-assisted authoring workflow in a scene representation and allows users to iteratively combine generated content with manual editing. These systems shift the role of the environment from a fixed setting for agent behavior toward a mutable context for human–AI authoring.

VsensAgent builds on this shift while applying it to virtual sensor data authoring. Its embodiment is functional and mediated through the simulation environment: it observes structured scene state and visual context, then acts through explicit skills. Users can directly modify the same environment, maintaining a shared state across conversational and manual actions. This configuration is both an editable artifact and the basis for generating sensor data. The simulation therefore serves simultaneously as an agent environment, a shared authoring workspace, and a data-generation substrate.

2.2 Human-Agent Interfaces for Complex Workflows

Human–agent collaboration has long been framed not as a choice between automation and direct manipulation, but as a negotiation of initiative. Mixed-initiative interfaces seek to provide computational assistance while preserving users' ability to direct, correct, or dismiss automated system actions [2, 11]. Context-aware task assistants operationalize this negotiation by tracking procedural state while allowing users to resolve uncertainty or reshape assistance through dialogue [3–5]. As agents move from recommending actions to executing them through external tools, the question becomes not only when they should intervene, but what parts of a workflow they may change and how users can maintain control.

This problem is amplified in stateful authoring, where intent develops alongside an intermediate artifact that preserves earlier decisions and constrains subsequent ones. Recent HCI systems support this process by decomposing model operations, exposing intermediate outputs, and coupling language with editable representations [33, 45]. Work in procedural and co-creative domains similarly treats AI-produced artifacts as material that users or practitioners can inspect and refine [12, 31]. Recent agentic development tools also connect conversational requests to persistent workspaces. Claude Code supports codebase inspection, file editing, command execution, and review of changes [32]. Related frameworks extends agent integration into spatial environments [24], while infrastructural tools such as Unity CLI [2], exposes commands enabling runtime intervention. *VsensAgent* shares these workflows' use of workspace context, tool-mediated execution, and user

[2]https://docs.unity.com/en-us/unity-cli/unity-cli

revision of intermediate results. In sensing authoring, these patterns support decisions about both what to observe and how to configure the scene to observe it.

Our study examines how users coordinate sensing decisions through operation-level handoffs in this shared scene, where changes to sensors, avatars, and motions also change the data produced

2.3 Virtual Sensor Data Generation and Authoring

Research on synthetic data for HAR has largely addressed data scarcity by expanding the signals available for model development. Existing approaches include signal-level generation and augmentation [1, 20, 26], cross-modal pipelines that derive virtual sensor signals from video, language, or synthesized animations [18, 22, 38], and physics-based simulation that exposes controllable motion and sensor parameters [15, 49]. These approaches reduce dependence on physical collection, but are typically evaluated through signal fidelity, dataset diversity, or downstream recognition performance. Decisions about what data to generate are consequently treated primarily as inputs to a technical pipeline.

Recent work has raised the abstraction of these pipelines. IMUGPT [22] uses language-driven motion generation to produce virtual IMU data, while AgentSense [23] employs LLM-guided embodied agents to generate personas and daily routines within simulated homes. These systems make synthetic data generation more expressive and scalable, but their agents principally automate the production of data; sustained user interaction with the intermediate sensing configuration is not the central object of study.

Complementary interface research has begun to treat configuration itself as an authoring problem. For example, prior work has explore the use of XR environment as authoring medium [28, 29]. Our focus is on how partially specified sensing goals become executable configurations and how authoring operations are distributed between user and agent.

3 Formative Study

To ground the design of *VsensAgent* in current authoring practices, we conducted formative interviews with four practitioners who had experience generating virtual sensor data for HAR. We examined where their workflows broke down, how sensing goals were translated into executable configurations, and where interactive agent support might be useful. The study aimed to identify design opportunities rather than characterize virtual sensing practice at large.

3.1 Methods

3.1.1 Participants. We recruited four practitioners (E1–E4), all male, aged 26–38 years ($M = 31.0$, $SD = 5.29$). All had backgrounds in computer science or engineering, experience developing HAR systems, and had completed at least one project involving virtual sensing. Participants were not compensated.

3.1.2 Interviews and Analysis. We conducted a semi-structured interview with each participant, lasting approximately 30 minutes. Interviews were audio-recorded with consent. We first asked participants to describe their virtual sensing workflows and the difficulties involved in configuring sensors, generating and inspecting signals, validating results, and iterating on datasets. We also asked about challenges they had observed when supporting less-experienced collaborators. Only after discussing existing practices did we invite participants to reflect on where agent assistance might or might not be appropriate. Following a thematic analysis process [34], two authors reviewed the transcripts, developed initial codes, and iteratively organized them into three themes.

3.2 Findings

3.2.1 Theme 1: Practical Knowledge Was Distributed and Often Implicit. Virtual sensor authoring required participants to coordinate conceptual knowledge about HAR with implementation details distributed across papers, code, 3D editors, and analysis tools. This created difficulties beyond learning any single interface. Participants noted that novices might understand the target activity but overlook how placement, orientation, and coordinate frames shape the resulting signal. Experienced practitioners had acquired much of this knowledge through trial and error. As E1 explained, "*There was no clear recipe. I had to look at papers, code, and examples, and still many details were implicit.*" Authoring therefore involved repeatedly translating scattered knowledge into concrete configuration and execution steps.

3.2.2 Theme 2: Sensing Intent Became Concrete Through Executable Configurations. Participants described a gap between how they formulated sensing problems and how existing tools required them to specify a solution. They typically began with activities, body regions, or desired distinctions rather than sensor coordinates and motion parameters. As E2 noted, "*We usually start from the question of motion difference, not from the exact coordinates of the sensor.*" Such goals did not always determine a unique configuration. Decisions about placement, orientation, motion variation, and recording conditions often became clearer only after participants could inspect a scene or generated signal. Translating intent was therefore not a one-time conversion from language to parameters, but a process of making partial goals concrete through execution.

3.2.3 Theme 3: Exploration Required Editable Configurations and Comparative Evidence. Participants valued simulation because it should make alternative sensing configurations inexpensive to test. In practice, however, even local changes could require returning to scripts or rebuilding parts of a pipeline. As E2 observed, "*Virtual data should be easy to change, but in reality every change still means going back to scripts or rebuilding part of the pipeline.*" Participants also found it difficult to preserve the relationship between a configuration and its results when comparing alternatives. E3 explained, "*I can generate another version, but then I need to remember what I changed and compare the results myself.*" Exploration consequently required more than producing additional samples: it required configurations that could be inspected and revised, together with evidence that could inform the next change.

Together, these themes yielded three design considerations for *VsensAgent*: connect contextual guidance to executable authoring operations; ground agent actions in the same evolving scene that users inspect and manipulate; and preserve an iterative loop between configuration, recording, analysis, and revision.

4 VsensAgent

VsensAgent is a human–agent interface that brings conversation, direct manipulation, simulation, and sensor-data analysis into a shared 3D authoring environment. Its central design premise is that the scene is not merely an output produced by the agent, but an executable artifact that both user and agent can inspect and modify as intent develops.

4.1 System Overview and Authoring Workflow

The interface pairs a conversational input with an editable Unity scene. Users can communicate through text or speech to describe sensing goals, ask questions, respond to clarifications, review proposals, and delegate operations. They may also directly manipulate sensors, avatars, and scene objects, control motion playback, and record virtual sensor data. The agent responds through text or synthesized audio and can configure sensors, place avatars, generate or load motions, control

scene objects, and initiate data analysis. Both interaction modes operate on the same scene: an agent-generated configuration remains directly editable, while user modifications become context for subsequent agent actions.

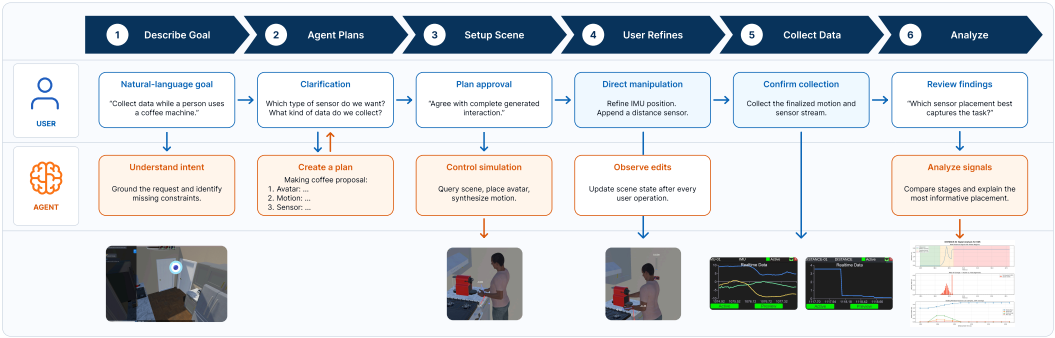


Fig. 1. An example *VsensAgent* workflow. The agent turns an initial sensing goal into an executable scene, the user directly refines the shared configuration, and recorded data supports subsequent analysis and revision.

Figure 1 illustrates one possible workflow. The user begins with a goal such as collecting sensor data while a person uses a coffee machine. The agent identifies missing constraints and proposes an initial plan. After confirmation, it constructs an executable scene from which the user can continue authoring. In this example, the user directly adjusts the IMU placement and adds a distance sensor before recording the resulting motion and sensor streams. The recorded data can then be analyzed and related back to the current configuration, providing evidence for further revision. This sequence is illustrative rather than prescribed: users may delegate, edit, record, or revisit earlier decisions at any point in the process.

Figure 2 organizes the system around three coupled entities: the *user*, the *agent*, and the *scene*. The user and agent communicate to develop sensing goals and coordinate actions. The scene provides their shared authoring state. The agent queries this state before executing scene-level actions and observes changes introduced through direct manipulation. When a request requires isolated or longer-running processing, the main agent can delegate the task to a subagent and return its result to the ongoing interaction.

These interactions form three connected loops. The *user-communication loop* supports goal specification, clarification, proposal selection, and response generation. The *scene-context observation and control loop* allows the user and agent to observe and modify the same executable configuration. The *delegate-task loop* supports sandboxed reasoning, code execution, and sensor-data analysis without granting the subagent direct control of the scene. These loops describe the functional architecture; the Python backend implements the agent workflow as procedural stages with conditional branches, rather than an explicit state-graph model. Each user turn updates scene context, invokes the model, optionally queries additional scene information for a follow-up model call, and dispatches structured actions. Applicable validation routines check the proposed operations, and the interaction is recorded before the response is returned. This orchestration governs individual agent turns without prescribing the user's authoring sequence.

4.2 Maintaining Shared Scene State

Natural-language history alone does not capture the state of a spatial authoring task. The configuration of the shared scene may affect what action is appropriate next. Moreover, users can directly edit the scene without describing those changes to the agent. *VsensAgent* therefore treats

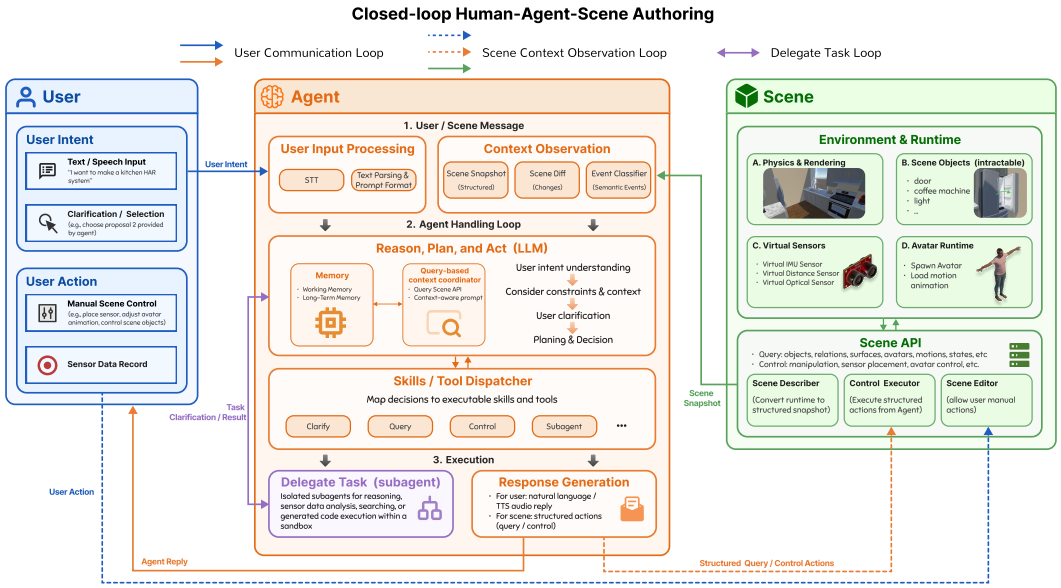


Fig. 2. *VsensAgent* couples user communication, shared-scene observation and control, and delegated-task execution. User and agent actions update the same executable scene, whose state and recorded data provide context for subsequent interaction.

the Unity runtime, rather than the conversation, as the authoritative representation of the current configuration.

The shared state is the live Unity scene that both interaction modes modify; a snapshot is a time-specific JSON description of its exposed objects and properties. Snapshots accompanying user messages and heartbeats are keyed by object name and contain spatial information, capability properties, current and available states, and, where applicable, sensor readings and attachment references. On-demand Scene API queries provide additional structured context, including object identifiers, scene versions, surfaces, zones, and relations. These representations allow the agent to resolve targets and formulate operations against the shared scene. When structured metadata is insufficient—for example, when checking a visually ambiguous placement or occlusion—the agent can autonomously request a screenshot for multimodal inspection [37].

To track changes between requests, *VsensAgent* receives scene snapshots through periodic heartbeats. For each heartbeat, the backend compares the incoming snapshot with its previous snapshot to compute a selective diff: object additions and removals, object-state changes, sensor-value changes subject to thresholds, and user-position changes. A rule-based event classifier maps these change records to `object-added`, `object-removed`, `state-change`, `sensor-trigger`, `sensor-anomaly`, or `user-movement events`; changes that do not meet the rules produce no event. Events are retained as context, while those meeting a predefined priority threshold trigger an agent call after cooldown and suppression of agent-originated changes. The resulting prompt supplies the event summary and current scene, allowing the agent to respond, invoke skills, or remain silent. The diff is not an exhaustive edit history: changes such as ordinary object rotation or attachment are available in the updated snapshot but are not individually reported by this diff. Thus, snapshots describe what is currently present, whereas diffs summarize selected changes between observations.

VsensAgent separates current working context from retrievable knowledge and interaction history. Working memory maintains the current scene state, active plan, and recent events. Prompt materials and domain knowledge are preloaded into a server-side vector database store for retrieval, while conversation and state histories are retained in searchable `.log` files. Across turns, the agent combines the updated working context with relevant retrieved material, rather than appending the entire log history to every prompt. This design limits context growth while preserving access to earlier interactions. In the user study,the preloaded knowledge remained available, but session interactions were not written back to the vector database as accumulated experience. Memory supports continuity but does not replace observation of the current scene. For example, when a user asks to "*try another placement*," the agent can compare the current snapshot with an earlier configuration while ensuring that its next action is applied to the scene as it now exists.

Together, these mechanisms preserve a common reference for subsequent interaction as either the user or the agent modifies the scene.

4.3 Executable Authoring Skills

Scene grounding provides the agent with an up-to-date representation of the authoring state; skills provide bounded means for changing that state. *VsensAgent* uses `gpt-5.6-sol` as its multimodal main model for language-based planning and visual inspection of screenshots. Following prior skill-based agent architectures [14, 51], the system exposes a registry of structured skills, each defined by an operation, input schema, execution endpoint, and return format. Rather than receiving unrestricted access to the Unity runtime, the model can affect the environment only through these registered capabilities. Agent skills are summarized in Figure 3

From intent to bounded execution. When a request requires an environment-level operation, the main agent selects one or more registered skills and supplies their required arguments. A dispatcher validates each call, routes it to the corresponding Unity or backend endpoint, and returns a structured result describing its status and affected objects. The main agent can then explain the outcome, continue with another operation, revise a failed call, or request clarification. Additional operations can be incorporated by registering a corresponding schema and execution endpoint with the dispatcher.

A request may therefore expand into several coordinated calls. For example, "*place an IMU on the avatar's wrist for this motion*" requires the agent to query the active avatar, resolve the wrist joint, create and attach the sensor, configure its offset and orientation, and return the resulting state for inspection. Intermediate results remain visible and directly editable by the user.

During this process, *VsensAgent*'s scene editing is asynchronous with respect to the Unity interface: users can continue manipulating the scene while the agent processes a request. Individual edit operations return completion results, but an agent turn does not lock the scene. The agent begins from the state available when the request is issued and incorporates intervening changes through subsequent heartbeat updates and scene queries. Coordination therefore relies on renewed observation and follow-up interaction, allowing users and the agent to retain initiative without requiring their activities to proceed in lockstep.

Authoring operations. The registry groups skills according to their role in the authoring process. Context and coordination skills support clarification, permission requests, scene queries, memory retrieval, and delegated-task management. The agent can submit, monitor, or cancel delegated tasks, while scene queries and memory retrieval reuse the shared-state mechanisms described in Section 4.2 rather than maintaining a separate representation of the environment.

Environment-construction skills operate on scene objects, virtual sensors, and avatars. Object skills modify editable object states and transforms, and can highlight relevant targets for user

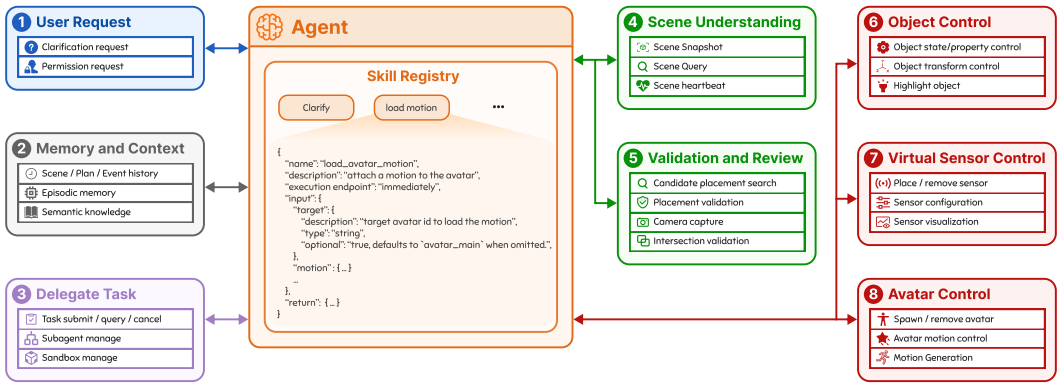


Fig. 3. *VsensAgent*'s skill registry. Skills connect the main agent to context and coordination, environment construction, and validation and review. Scene changes are mediated through structured calls, while data-level review is delegated to an analysis subagent.

attention. Sensor skills create, remove, attach, orient, configure, and visualize virtual sensors, and can initiate labeled recordings from the active configuration. In the study, *VsensAgent* supported virtual IMU and distance sensors. Both user- and agent-initiated recordings were stored with their sensor identities, timestamps, and activity labels for subsequent inspection and analysis.

Avatar skills spawn, remove, and position avatars, load existing motion clips, control playback, and invoke motion generation. When no suitable motion is available, the main agent expands the user's activity description into a scene-grounded prompt and submits it to `HY-Motion-1.0` [43]. Server-side services will return the animation sequence, convert it into the runtime format, after which it can be applied on the avatar.

Validation and review. Validation and review operate at two distinct levels, while neither of them determines the final design autonomously: configurations remain subject to user judgment.. At the scene level, deterministic checks identify geometric construction problems, including invalid attachment, object intersection, floating placement, and the absence of a feasible candidate surface. Their structured results allow the main agent to adjust an operation, propose alternatives, or return the decision to the user.

At the data level, review skills can delegate selected recordings to a subagent powered by `claude-sonnet-4-6`. The subagent receives the sensing goal, relevant scene context, recording labels, and sensor streams, and operates in a server-side sandbox where it can execute analysis code, compare recordings or configurations, and produce visualizations and a written report. The report is returned to both the user and the main agent, which may use the findings to discuss possible revisions to the current configuration. The subagent does not modify the scene directly, and its analysis remains advisory.

4.4 Implementation Details

VsensAgent was implemented as a Unity 6 client and a Python backend connected through a bidirectional WebSocket interface that transmits scene observations, structured skill calls, and execution results. The main agent can autonomously query scene snapshots and request screenshots as needed during its reasoning loop, while LanceDB [3] stores preloaded prompt materials and domain knowledge on the server for retrieval. Virtual sensors are sampled at 60 Hz: IMU signals are derived

[3]https://www.lancedb.com/

directly from simulated sensor motion without additional noise or filtering [38], and distance sensors use ray casting with a sufficient range of 20 m [48].

5 User Study

We conducted a remote task-based study to examine how people collaborate with *VsensAgent*. Participants completed three tasks with different sensing constraints, allowing us to trace how intent became executable, how control moved between language and direct manipulation, and how participants experienced this form of assistance. We address the following research questions:

- **RQ1.** How do users progressively articulate and operationalize authoring intent through interaction with the agent and the shared scene?
- **RQ2.** How do users distribute and coordinate authoring operations between natural-language delegation and direct scene manipulation?
- **RQ3.** What opportunities and tensions does agent-assisted virtual sensing reveal for human–agent interfaces in complex, stateful authoring tasks?

5.1 Study Setup and Tasks

Figure 4(a) shows the study interface. In addition to interacting with the agent, participants could navigate the scene, switch to a bird's-eye view, manipulate objects and sensors with transform gizmos, and control recording and data review. An animated floating eye visualizes the agent's current status. These controls exposed the authoring capabilities described in Section 4.

All tasks took place in the same virtual kitchen, containing a stove, sink, preparation table, cabinets, coffee machine, and other editable objects (Figure 4(b)). Its compact layout kept the authoring scope bounded and inspectable, while its functionally distinct objects provided meaningful targets for the sensing tasks and avatar motions. Holding the environment constant supported comparison across tasks, which varied the source of design uncertainty: allocating limited sensors, choosing a wearable representation, and developing a strategy for an under-specified sensing objective.

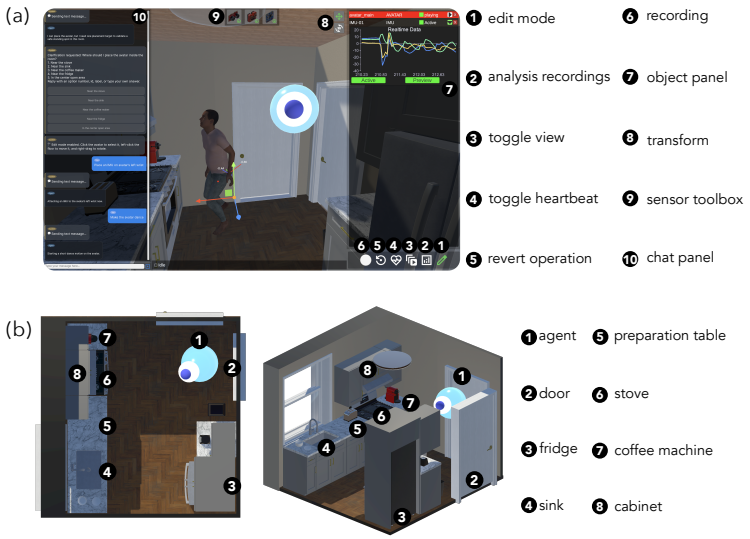


Fig. 4. Study environment. (a) The interface supports agent interaction, scene navigation, direct editing, and sensor recording. (b) The virtual kitchen used across all three tasks.

Task 1: Sensor allocation under resource constraints. Participants designed a configuration that could distinguish *cooking at the stove*, *opening the cabinet above the stove*, and *washing hands at the sink*. They could use at most two virtual IMU or distance sensors; modality, placement, and orientation remained open.

Task 2: Wearable IMU placement. Participants used a single body-mounted IMU to distinguish *chopping at the preparation table*, *washing hands at the sink*, and *making coffee*. They determined where and how the IMU should be attached to obtain a useful motion representation across the three activities.

Task 3: Rough user localization. Participants designed a configuration for estimating a user's approximate location in the kitchen without identifying the activity being performed. They could use up to four virtual IMU or distance sensors. Because neither the relevant regions nor the sensing strategy was predefined, participants had to decide how location should be represented as well as how it should be sensed.

For all tasks, participants were instructed to keep placements physically plausible—for example, attached or supported rather than visibly floating or intersecting other objects. After completing each configuration, they recorded at least 10 seconds of sensor data for each requested activity or tested condition.

5.2 Participants

ID	gender	age	occup.	exp. In HAR	3D/ simulation use	AI use	background
P1	Male	33	Faculty	Yes	Frequently	Frequently	HCI, AI, Ubicomp, Robotics
P2	Female	30	Ph.D student	Yes	Rarely	Frequently	HCI, Robotics
P3	Male	28	Ph.D student	Yes	Occasionally	Occasionally	HCI, Ubicomp,
P4	Female	25	Ph.D student	No	Frequently	Frequently	HCI
P5	Female	34	Researcher	Yes	Rarely	Rarely	HCI, Ubicomp
P6	Female	27	Master's Student	No	Never	Frequently	Healthcare
P7	Male	29	Ph.D student	Yes	Frequently	Frequently	HCI
P8	Male	31	Faculty	Yes	Never	Frequently	HCI, Ubicomp, Healthcare
P9	Male	22	Master's Student	Yes	Occasionally	Frequently	HCI, AI
P10	Male	22	Master's Student	Yes	Frequently	Occasionally	HCI, Ubicomp
P11	Male	25	Master's Student	Yes	Frequently	Frequently	HCI, AI, Ubicomp
P12	Male	23	Master's Student	Yes	Frequently	Frequently	HCI, AI, Ubicomp

Fig. 5. Participants' professional backgrounds and self-reported experience with sensor-based systems, 3D or simulation tools, and AI-based tools. Here, "AI use" refers specifically to experience with agentic coding and co-work platforms, rather than casual conversational use of AI chatbots.

We recruited 12 participants (8 men and 4 women) from universities, aged 22–34 years ($M = 27.4$, $SD = 4.1$); none had participated in the formative study. The sample included faculty members, a researcher, doctoral students, and master's students, primarily from HCI and ubiquitous computing, with additional backgrounds in artificial intelligence, robotics, and healthcare. Ten participants had previously developed at least one sensor-based system. As summarized in Figure 5, their reported use of 3D or simulation environments and AI-based tools ranged from none or rare use to frequent use.

5.3 Procedure and Data Collection

Participants received a distributable build of *VsensAgent*, a written guide, and a demonstration video. Before the formal tasks, they could freely test the interface and the agent in a separate practice scene. During the tasks, they could communicate with the agent in any language and combine natural-language delegation with direct scene manipulation. The researchers provided no task-specific guidance.

Participants completed the tasks independently and in a fixed order, with the application restarted between tasks. The fixed order standardized participants' prior exposure to the interface before they encountered Task 3, the most under-specified task. Scene state, conversation history, and working memory were cleared between tasks.

The sessions were self-paced. The system recorded timestamped user and agent messages, agent-executed operations, manual scene edits, system events, data-recording actions, and final sensing configurations. Participants also recorded their screens, which we used to resolve spatial actions or interface states that could not be reconstructed unambiguously from the logs.

After completing the tasks, each participant joined an approximately 40-minute remote, semi-structured interview. Using a dedicated log visualization, the interviewer and participant revisited selected episodes and discussed the participant's decisions, expectations, and experience with the agent. Interviews were audio-recorded and transcribed. The study received institutional ethics approval; all participants provided informed consent and were compensated at a rate equivalent to USD 10 per hour.

6 Results

We report findings from interaction logs, authored configurations, screen recordings, and log-elicited interviews. Because each research question required a different analytical lens, we introduce the corresponding coding and aggregation procedures alongside the relevant findings. Our analysis is exploratory and emphasizes recurring patterns and contrasts rather than statistical hypothesis testing.

6.1 Overview of Authoring Activity

Across the 36 task sessions, participants spent an average of 19 min 46 s per task (median = 12 min 33 s), with substantial variation across both participants and tasks. Completion of all three tasks took 59 min 19 s on average (median = 46 min 56 s; range = 18 min 55 s–117 min 09 s). Task 2 was generally completed more quickly than the other tasks, whereas Task 3 showed the widest spread in completion time: although its median duration was 11 min 39 s, several participants engaged in substantially longer sessions, resulting in a mean of 22 min 32 s. A similar pattern appeared in operation counts. Participants performed an average of 65.3 logged operations across the three tasks, with Task 3 involving the largest number of operations on average (M = 29, median = 26), compared with Task 1 (M = 21.8, median = 20) and Task 2 (M = 14.5, median = 12).

The normalized timelines in Figure 6 show how participants combined agent interactions, manual scene edits, and sensor-recording within each session. The timelines reveal considerable variation in how participants organized the authoring process. Some sessions were dominated by extended exchanges with the agent (e.g. P4 and P5), while others alternated frequently between agent-mediated actions and direct manipulation (e.g. P1 and P9). Manual editing appeared both as an early means of establishing a configuration and as a later intervention following agent-generated actions. Recording was generally concentrated toward the end of a session, although several participants recorded intermediate configurations before continuing to revise the scene. These patterns indicate that participants did not follow a uniform authoring sequence.

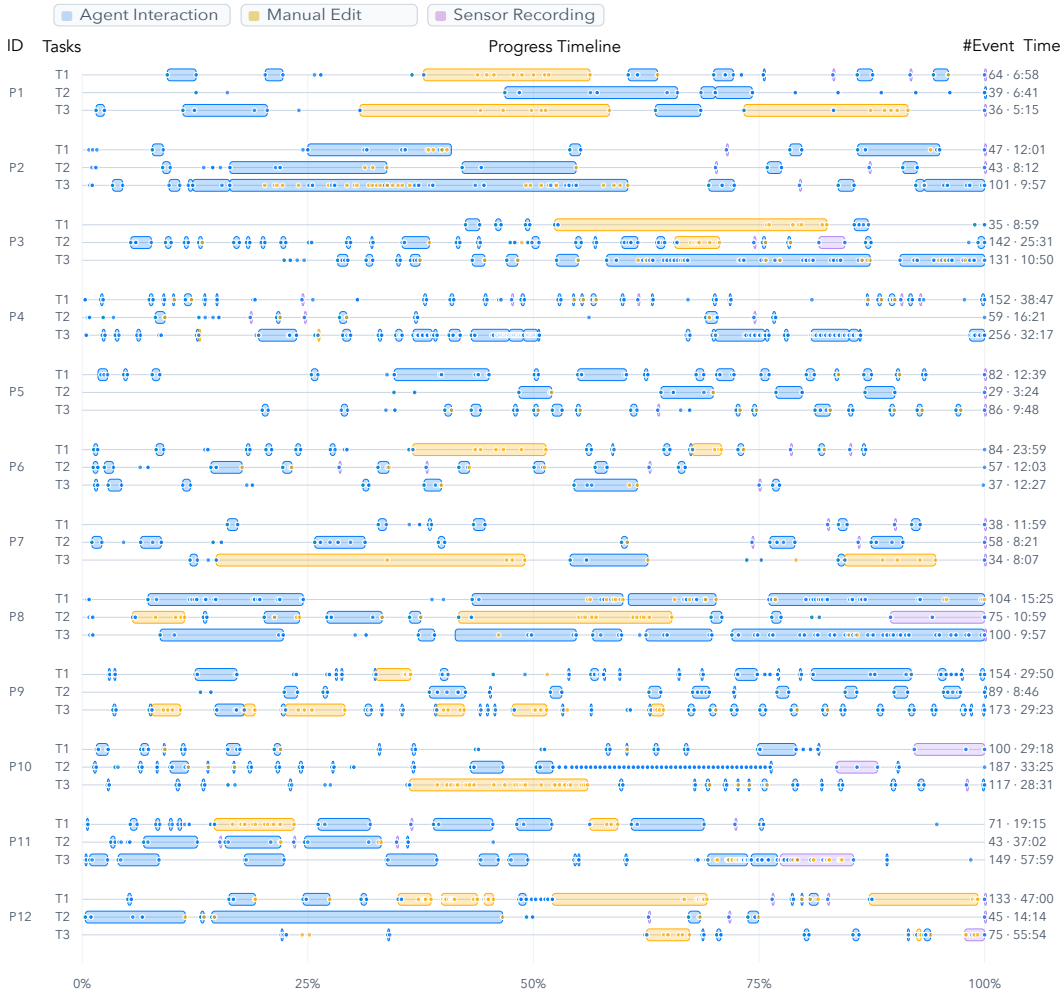


Fig. 6. Normalized timelines of agent interactions, manual scene edits, and sensor recordings across the 36 task sessions.

To characterize natural-language interaction within these authoring trajectories, two researchers independently coded each user message by its primary interaction act: the response or action expected from the agent next (Figure 7). The coding scheme and precedence rules were established before independent coding. Although each message was the unit of analysis, coders considered adjacent dialogue turns and screen recordings to interpret its role in the ongoing authoring process. Ambiguous cases and coding disagreements were resolved through discussion using this context. The coding scheme included eight substantive categories:

- *Framing*, for introducing or reframing the sensing task;
- *Inquiry*, for requesting information, explanations, system state, or task-related procedural guidance;
- *Planning*, for soliciting recommendations, alternatives, comparisons, or rationale before committing to a design;



Fig. 7. Normalized timelines of user utterances according to their primary interaction acts.

- *Specification*, for providing constraints, preferences, positions, orientations, or other design parameters;
- *Confirmation*, for accepting, rejecting, or selecting an existing proposal;
- *Execution*, for requesting an operation such as creating, placing, moving, deleting, or recording;
- *Evaluation*, for asking the agent to assess or verify an existing result without explicitly requesting a change;
- *Repair*, for identifying a problem and requesting correction, refinement, retry, or recovery.

Each utterance received exactly one primary label. For utterances containing multiple acts, we assigned the label corresponding to the requested next response, using the following precedence: *repair*, *execution*, *confirmation*, *evaluation*, *planning*, *inquiry*, *specification*, and *framing*. Greetings, acknowledgments, typos, filler, and utterances that did not fit these categories were coded as *Other*.

For repair utterances, we additionally distinguished correction of a misunderstanding, refinement of an otherwise valid result, retry of an unsuccessful action, and recovery through undoing or restoring state. Target entities, requested actions, and explicit constraints were annotated separately from the interaction-act label.

Based on usage costs reported by the respective providers' API platforms, the average API cost across all processing stages was approximately USD 5.1 per participant for the three tasks combined. Motion generation model was deployed locally hence incurred no API charges.

6.2 From Prescribed Tasks to Authoring Intent

This section examines how participants moved from an initial interpretation of each task into execution, and how the resulting scene states supported further specification.

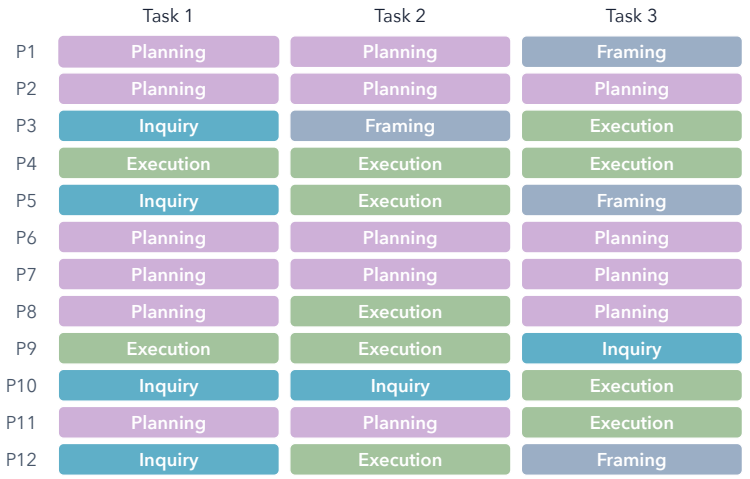


Fig. 8. Participants' initial interaction strategies across the three tasks.

6.2.1 Diverse Entry Strategies, Rapid Transition to Action. We defined the initial strategy as the first user utterance not coded as *other*. As shown in Figure 8, planning was the most common entry in Task 1 (6/12), while Task 2 was evenly split between planning and execution (5/12 each); Task 3 showed the most heterogeneous pattern. Task structure may have shaped, but did not determine, how participants entered authoring.

Despite these differences, participants moved into execution quickly. The median number of user turns before the first execution request was one in all three tasks. Participants mentioned a median of one configuration topic before execution. First execution requests thus typically preceded a complete specification.

6.2.2 Configuration Details Emerged Around the First Execution. To examine what became specified at each stage, we regrouped the separately annotated targets and constraints from Section 6.1 into eight content-oriented configuration topics, shown in Figure 9. Because the phases contained different numbers of turns, the comparison indicates when a topic appeared, not its relative frequency.

Before execution, participants primarily discussed sensing goals and resource constraints. At or after the first execution request, placement, orientation, avatar and motion scenarios, recording protocols, and evaluation criteria became more visible in the interaction. In Task 3, goal and area

	Before first execution			At / after first execution		
	Task 1	Task 2	Task 3	Task 1	Task 2	Task 3
Goal / activity / area definition	3/12	2/12	3/12	3/12	6/12	6/12
Sensor count / type	5/12	4/12	6/12	4/12	7/12	7/12
Placement / body mounting	2/12	2/12	0/12	10/12	10/12	9/12
Orientation / coverage	1/12	0/12	1/12	4/12	4/12	2/12
Avatar / motion / scenario	1/12	2/12	0/12	10/12	11/12	11/12
Recording protocol	0/12	0/12	0/12	6/12	5/12	4/12
Evaluation / analysis criteria	0/12	0/12	0/12	1/12	4/12	1/12
Other configuration	1/12	0/12	1/12	4/12	3/12	4/12

Fig. 9. Configuration topics mentioned before and at or after the first execution request. Each cell reports the number of sessions, out of 12 per task, in which the topic appeared.

definitions also continued into this phase because the spatial representation itself remained open. The first execution request therefore marked a practical transition from introducing the sensing problem to progressively working out its implementation.

6.2.3 Executable States as Shared References. Figure 10 illustrates how the resulting scene states grounded subsequent interaction. In P1–T1, the participant manually revised the sensing configuration before asking the agent to place an avatar and record a cabinet-opening motion. In P4–T3, the participant first placed sensors on walls facing the room center, then aligned them with the avatar's chest height. In P6–T1, the participant asked the agent to assess a sensor inside the cabinet and authorized it to adjust the placement.

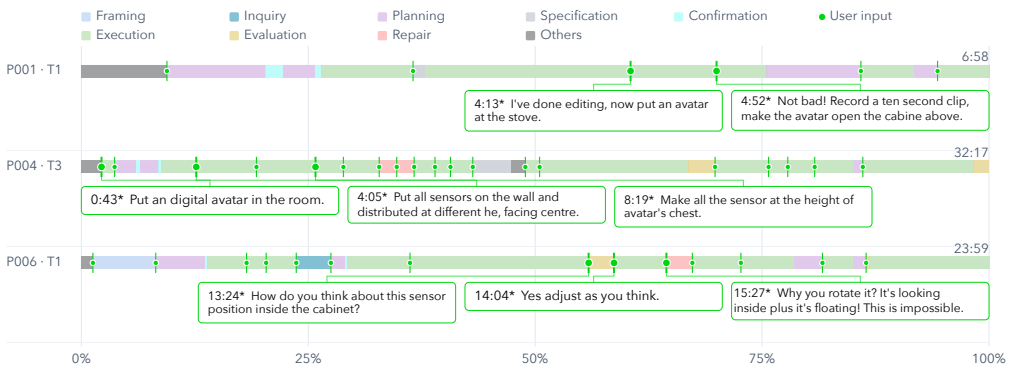


Fig. 10. Representative sequences in which participants used executable scene states as shared references.

Together, these cases show that the scene carried context across turns and interaction modalities. Participants could refer to existing objects, add constraints, and request evaluation or repair without restating the complete configuration. Executable states therefore served not only as outcomes of prior instructions, but also as shared references through which subsequent intent could be expressed.

6.3 Delegation and Direct Manipulation Across Operations

Participants could either delegate operations through natural language or manipulate the scene directly. To avoid overcounting continuous interface updates, we grouped consecutive scene events involving the same actor, target, and operation into manipulation episodes. Figure 11(a) compares the shares of user- and agent-performed episodes, whereas Figure 11(b) characterizes the operations requested through delegation.

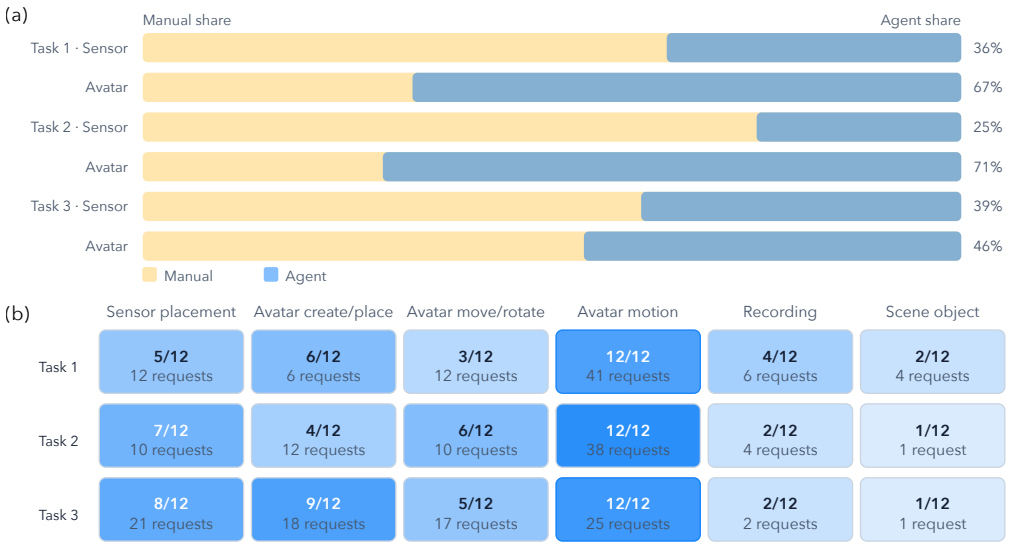


Fig. 11. Distribution and content of authoring operations. (a) Shares of sensor- and avatar-manipulation episodes performed manually or by the agent in each task. (b) Operations requested through delegation; each $n/12$ value reports the number of participants who issued at least one corresponding request, followed by the total number of requests.

Control differs by operation rather than by object alone. All participants used the agent for avatar-motion generation in every task, reflecting the available interaction path rather than a demonstrated preference for delegation. Sensor editing showed the opposite aggregate pattern, with manual episodes accounting for a relatively large share across all three tasks. Yet participants still delegated sensor placement, particularly in Task 3 (8/12 participants; 21 requests). The same sensing object could therefore involve delegation at the level of a placement request and direct manipulation at the level of geometric adjustment.

Task 3 also involved the broadest use of delegated avatar creation and placement (9/12 participants; 18 requests). These task-level contrasts were accompanied by substantial variation between participants. As Figure 6 shows, manual editing was concentrated in particular participants and sessions rather than distributed uniformly; participants allocated control locally according to the operation, task, and their own authoring practices.

6.4 Exploring an Under-specified Sensing Design Space

Task 3 left both the representation of rough location and its sensing configuration open. We therefore examined how participants translated this under-specified objective into end-of-session layouts, and how these layouts related to the agent's initial recommendations.

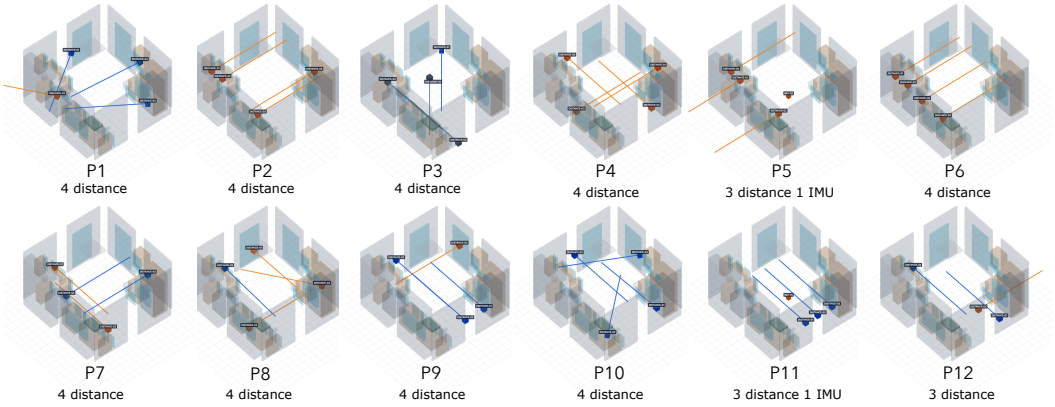


Fig. 12. Sensor layout overview of 12 participants in Task 3. The orange color indicates the sensor is placed by agent, while the blue color represents sensor was adjusted by user. Colored points with raycast show the direction of distance sensor, while the isolated points are IMU sensors (e.g. P5 and P11).

6.4.1 Diverse Interpretations of Rough Localization. All 12 used distance sensors, 11 used the full four-sensor budget, and only two incorporated an IMU (Figure 12). Most sensors were also positioned near torso height. Despite the open-ended objective, final configurations varied little in sensor modality or quantity.

Variation instead appeared in how participants encoded location spatially. Some organized sensing around functional landmarks, assigning beams to areas such as the stove, sink, coffee machine, and refrigerator. Others treated the kitchen as a geometric space to be partitioned by beams crossing walls, standing areas, or transition paths. A smaller number reasoned about location through the joint pattern of several sensor readings or supplemented spatial coverage with an IMU. Participants thus converged on the sensing components while differing in the spatial representation those components were intended to realize.

6.4.2 From Initial Proposals to Final Configurations. The convergence in final configurations was consistent with the solution frame recurring in the agent's initial proposals. Across comparable initial proposals, the agent consistently favored distance-based sensing and usually recommended using four sensors organized around functional zones. End-of-session layouts commonly retained these high-level choices, but the proposed spatial arrangements were not necessarily accepted as final.

Participants repositioned sensors, redirected their coverage, redefined the target regions, and, in some cases, changed the modality composition or reduced the number of sensors. In Figure 13, for example, P4 retained four distance sensors but reorganized them across the room's walls, whereas P12 rejected redundant and occluded placements and reframed the configuration around users' standing areas. Final designs often shared the broad strategy of the initial proposals, while participants revised and sometimes challenged specific design choices. The correspondence between proposals and outcomes suggests a common design scaffold, although it does not isolate agent influence from the affordances of the task itself.

6.5 Semi-structured Interviews

After the tasks, we conducted log-elicited semi-structured interviews to contextualize these findings and discuss participants' design decisions. Interviews were transcribed using automatic speech

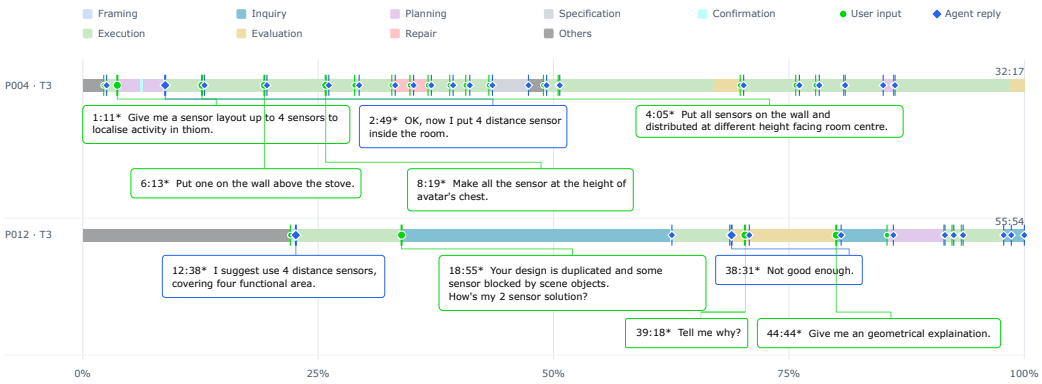


Fig. 13. Example conversation with P4 and P12 in Task 3. Longer agent responses are abbreviated for readability.

recognition and translated into English. An author proficient in the respective interview language checked the transcripts against the recordings and reviewed the translations for accuracy. Quoted excerpts were checked against the original-language context to preserve participants' intended meanings. Transcripts were analyzed using inductive thematic analysis by two researchers [34]: first open-coded by two researchers independently, followed by iterative clustering into higher-order themes. Coding disagreements were discussed and refined through consensus. After the analysis, 4 recurring themes emerged: (1) *Assistance at the Boundary of Existing Expertise*; (2) *Executable Proposals Supported Imagination and Revision*; (3) *Trust Developed Through Situated Demonstrations*; (4) *Natural Language Reduced Discovery Costs but Required Manual Refinement for Precision.*

6.5.1 Theme 1: Assistance at the Boundary of Existing Expertise. The value of the agent varied with participants' prior knowledge. For those participants with limited familiarity with sensors, for example P4 works on XR and P6 works in medical area, found the workflow understandable because the agent could supply missing sensing knowledge and translate the task into concrete operations.

> "*Mostly I rely on agent's proposals, which generally make sense.*" (P4); "*I do not know sensors very well, but I could still understand what was happening and rely on the AI to help me decide what to use.*" (P6).

For participants with extensive HAR experience, the agent served a different role. P3 and P11 often formed a sensing strategy as soon as they read the task, but described using the agent to save time when instantiating motions, placing objects, and coordinating the pipeline. They expected its knowledge to become more valuable when working with unfamiliar or more complex sensor types.

> "*Honestly I don't need a proposal to complete the tasks. If the sensor is something I'm unfamiliar with, I'll rely more on the AI's proposal... The agent mainly saved time.*" (P3); "*I used agent to assist and execute my own thoughts.*" (P11).

P8 similarly viewed the system as a tutorial and knowledge resource whose importance would grow as the available design space expanded. Agent assistance was therefore most valuable not to a fixed category of "novice" users, but at the shifting boundary of what each participant already knew.

6.5.2 Theme 2: Executable Proposals Supported Imagination and Revision. Most participants valued the agent's proposals as editable starting points rather than complete solutions. Participants described proposals as making it easier to begin an under-specified task by turning it into a concrete scene that participants could inspect and revise. For example, P2 and P5 described generating the avatar motion early in the process to help them imagine where the person would stand, how the activity would unfold, and what a sensor would need to capture.

> "*I wanted to see the motion first. Once the avatar performed it in the kitchen, it became easier to imagine where the sensors should go.*" (P2); "*The agent provide a first draft on the blank paper for me to edit.*" (P5).

This role of execution extended beyond automation: generated motions and configurations became resources for thinking through the sensing problem. Participants could identify missing constraints, notice occlusion or coverage problems, and refine their ideas in relation to an observable state. P11's request for "*more visualized initial proposals*" further suggests that users may benefit from several alternatives with different rationales, rather than a single authoritative recommendation.

6.5.3 Theme 3: Trust Developed Through Situated Demonstrations. Participants generally evaluated the agent's spatial understanding positively, despite occasional errors in placement or orientation. Their trust, however, was not established solely through its explanations. Half of the participants tested the agent's capabilities before beginning the prescribed tasks, using low-risk requests to determine what it could observe and manipulate. P5 and P9 subsequently reported that their confidence increased as the tasks progressed.

> "*...my trust on the agent progress with the tasks.*" (P5); "*At first I was testing what it could do, but after it understood the scene and completed several operations, I became more willing to rely on it.*" (P9).

Participants also discussed their confidence on the generated data. HAR-experienced participants considered the data quality acceptable: "*because beneath the agent interface, it's just a standard virtual sensor pipeline.*" (P7). P1 described the subagent's analysis function as "*...offering a window for inspection, definitely more convenient for having this integration.*". P1 acknowledged that the generated report was not entirely reliable but considered it sufficient for preliminary prototyping.

6.5.4 Theme 4: Natural Language Supported Discovery but Required Manual Refinement for Precision. Participants did not view natural language as the most efficient medium for every authoring operation. Participants familiar with 3D editing (e.g. P1, P11 and P12) noted that verbal interaction could be time-consuming and sometimes insufficient for communicating precise spatial intent. They described direct manipulation as faster when the desired position or orientation was already clear, and language as more useful for uncertain, unfamiliar, or multi-step operations that required discovering functions or learning procedures

> "*Imagine I'm designing a full smart home environment, like 100 square meters, it would be challenging to convey my intent...*" (P1); "*The agent seems knowledgeable on spatial position for each 3D object, but if I require a detailed positioning, it need to be refined manually.*" (P12).

P10 connected this role to familiar difficulties in CAD-like systems: as the number of functions grows, users must search for tools and learn how individual operations work. From this perspective, the agent can serve as a coordination and discovery layer over a complex authoring environment, while direct manipulation remains necessary for precise control. The two interaction modes were

therefore complementary, with their usefulness changing as intent moved from an open sensing problem toward a concrete spatial configuration.

Across the interviews, participants described reliance on the agent as conditional on their expertise, observed evidence of its competence, and access to direct control.

7 Discussion

Our findings connect progressive specification, shared control, and adpative assistance in virtual sensor authoring. We discuss three design implications grounded in these observations, with potential relevance to other stateful authoring workflows.

7.1 From Parameter Specification to Authoring Through Execution

Virtual sensor generation is often framed as a parameter-specification problem: users first determine the activity, sensors, placements, and motions, and then execute the resulting configuration [18, 22, 23]. In our study, participants often began with partial specifications and developed placement, motion, and recording constraints through inspection of the resulting scene.

Execution therefore served an *epistemic*, rather than merely productive, role [16]. An executable state externalized an otherwise underspecified intention: participants could point to it, evaluate it, modify it manually, or ask the agent to continue from it. This is consistent with accounts of situated action and reflection-in-action, in which plans are progressively elaborated through engagement with their consequences [39, 41]. In *VsensAgent*, the shared 3D scene made those consequences jointly available to both user and agent, turning execution into a medium for communication.

This shift also broadens what it means to author virtual sensor data. Participants were not simply selecting parameters for a fixed pipeline. They were constructing an *observation model*: deciding which activities and spatial relationships should become visible in the generated signals. The varied Task 3 configurations illustrate that the same sensing objective can be operationalized through multiple defensible scene arrangements. This extends accounts of provisional artifacts as resources for thinking [8, 40] to sensing configurations: participants revised which regions or activities to observe together with the placements and motions intended to make them observable. The persistent scene carried these decisions across subsequent manual edits and agent requests.

These findings motivate interfaces based on *progressive commitment*. Rather than requiring a complete prompt or autonomously inferring a final specification, an agent should create partial, inspectable, and reversible states that invite further refinement. Under this view, the agent's role is not to replace specification, but to help users make evolving intent executable, observable, and revisable.

7.2 Co-agency at the Granularity of Operations

If intent develops through execution, control cannot be allocated once for the entire authoring process. In *VsensAgent*, participants did not adopt a stable "manual" or "agent-driven" workflow. Instead, they repeatedly reassigned control across operations: the agent commonly generated avatar motions and initial configurations, while participants retained direct control over many spatial refinements. Delegation was therefore bounded and reversible rather than a sustained transfer of agency.

This operation-level allocation resonates with Parasuraman et al.'s argument that automation should be configured differently across information acquisition, analysis, decision selection, and action implementation, rather than treated as a single global level [35]. Mixed-initiative interaction recognizes that initiative may shift during a task [11]; in our study, these shifts often occurred at the level of individual operations. Language was effective for expressing semantic goals, coordinating

multiple objects, and invoking procedurally expensive actions. Direct manipulation was better suited to precise placement, rapid correction, and visually verifying local changes.

The strong delegation of avatar motion should not, however, be interpreted simply as a general preference for agent control. Motion synthesis was explicitly mediated by the agent, whereas geometric edits were readily available through direct manipulation. The observed division of labor thus reflects both user strategy and the distribution of system affordances. More broadly, this indicates that co-agency is shaped by the relative cost, precision, and inspectability of the available interaction techniques. Participants also developed different editing habits, suggesting that an effective allocation cannot be prescribed uniformly across users or tasks.

Designing for operation-level co-agency requires more than offering both a chat interface and a 3D editor. Actions performed through either modality must remain continuous in a shared state: users should be able to manually revise agent-generated content, and the agent should be able to interpret and build upon those revisions. Delegation should also be scoped to inspectable operations, with clear targets, visible effects, and lightweight correction. Rather than asking users to choose a global level of autonomy, authoring systems should make control easy to hand off, reclaim, and renegotiate as the artifact evolves.

7.3 Adaptive Scaffolding Without Collapsing the Design Space

The same agent assistance served different purposes across participants. For participants with limited sensing experience, the agent made the task approachable by supplying relevant concepts and an executable starting point. Experienced participants often already had a configuration in mind, but still used the agent to reduce setup effort or to access knowledge beyond their familiar sensor types. Agent support therefore did not simply compensate for a lack of expertise; it shifted between conceptual scaffolding, procedural assistance, and domain consultation according to what the user needed.

Participants valued executable proposals as starting points without requiring participants to accept a complete solution. Participants commonly treated the generated scene as a draft and revised its spatial details through further requests or direct manipulation. This suggests that effective scaffolding should be actionable but non-prescriptive: it should provide enough structure to enable progress while preserving clear opportunities to inspect, reject, and modify its assumptions.

Task 3 also reveals a potential tension. Final configurations converged around a common high-level strategy—typically four distance sensors—while differing substantially in their spatial realization. Participants' revisions and rejection of proposals demonstrate active evaluation, but do not establish whether the initial proposals influenced which alternatives they considered. This convergence may reflect task constraints, participants' prior sensing knowledge and established authoring practices, the agent's recurring proposals, or a combination of these factors. Our exploratory study of 12 participants, without a no-agent condition, cannot disentangle these influences or determine whether the agent narrowed the explored design space. Nevertheless, repeated initial proposals may create an anchoring risk, particularly when several plausible sensing strategies exist, as suggested by prior work on design fixation [13, 44].

Adaptive scaffolding should therefore regulate not only how much assistance is provided, but also how broadly the system represents the available alternatives. Instead of presenting one configuration as an implicit default, an agent could offer contrasting strategies, explain their assumptions and trade-offs, or invite users to articulate an initial approach before generating recommendations [42]. As users demonstrate greater confidence or take more direct control, the system could reduce explanatory detail while retaining access to deeper guidance when unfamiliar sensors or environments arise. The goal is not to maximize agent contribution, but to provide sufficient support without prematurely determining what counts as a valid sensing design.

8 Limitations & Future Work

Scope and ecological validity. Our study involved 12 participants, one kitchen environment, three bounded tasks, and a limited set of sensor modalities. This scope enabled close analysis of human–agent interaction, but does not capture the complexity of larger or longer-term sensing projects. Real environments could be imported through 3D scanning or Gaussian-splatting reconstruction, reducing the need to model each scene manually [54]. However, visual reconstruction alone does not provide the interactive objects, affordances, and structured scene knowledge required for agent reasoning and execution. Connecting captured environments to editable, semantically grounded scene representations therefore remains an important research challenge. In larger or multi-room settings, language may also need to be complemented by pointing, region selection, sketching, or other spatial input for communicating intent efficiently. Although we evaluated the system through a desktop interface, the same scene-grounded architecture could also be integrated into VR, where embodied manipulation, pointing, and spatial reference may produce different patterns of delegation and control.

Virtual sensor data validity. Our evaluation focused on collaborative authoring rather than sensor-data validity. The signal-generation procedures follow prior virtual-sensing methods [18, 28, 29, 38, 48, 53], providing a methodological basis for the simulation but not validating each participant-authored configuration. Participants independently chose placements, orientations, motions, and tested conditions, and the study did not prescribe a standardized protocol for assessing whether the resulting signals met each sensing objective. Future work could separate open-ended authoring from signal validation by evaluating authored configurations under standardized activities and layouts, comparing simulated and real signals, or measuring downstream model performance.

Causal and system-specific interpretation. We used a fixed task order to support progressive familiarization before participants undertook Task 3, the most under-specified task. Our aim was to characterize collaboration within this progression, rather than isolate the effects of task type. Nevertheless, task characteristics and accumulated experience cannot be separated: growing familiarity and trust may also have shaped participants' behavior in Task 3. Cross-task differences should therefore be interpreted within this ordered study context.

The study did not include editor-only, chat-only, or no-agent conditions. Our results therefore characterize how participants coordinated with *VsensAgent*, but do not establish improvements in efficiency or configuration quality, nor whether agent proposals caused convergence in design. Selecting a matched baseline is also non-trivial: an editor-only condition necessarily removes the proposal, knowledge-support, and operation-dispatch capabilities under study, whereas a chat-only condition removes the shared editable scene. Performance differences against either baseline would therefore bundle several interaction capabilities and offer limited causal explanation. Moreover, the observed division of labor reflects the current implementation, in which motion synthesis was primarily exposed through the agent and geometric refinement through direct manipulation. Future work could use capability-matched ablations to isolate scene grounding, proposal generation, and operation execution.

9 Conclusion

We presented *VsensAgent*, a human–agent interface that unifies language, direct manipulation, simulation, and analysis in a shared 3D scene. Across three tasks, 12 participants developed incomplete goals through execution and shifted between agent delegation and manual editing at the level of individual operations. These findings frame virtual sensor generation as progressive,

collaborative authoring in which agent-generated configurations remain inspectable and open to user revision.

Acknowledgments

This work was partially supported by JST Keio-SPRING (Grant Number Y01GQ25172, Y01GQ25160), JST PRESTO (Grant Number JPMJPR2134), JST AIP Acceleration Research (Grant Number JPMJCR25U4), and JSPS KAKENHI (Grant Number JP25K03168). Part of this work was supported by the Natural Science Basic Research Program of Shaanxi (Program No.2025JC-YBQN-852) and China Postdoctoral Science Foundation (No.2024M762548).

References

[1] Moustafa Farid Alzantot, Supriyo Chakraborty, and Mani B. Srivastava. 2017. SenseGen: A deep learning architecture for synthetic sensor data generation. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (2017), 188–193. https://api.semanticscholar.org/CorpusID:5159069

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300233

[3] Riku Arakawa, Prasoon Patidar, Will Page, Jill Lehman, and Mayank Goel. 2025. Scaling Context-Aware Task Assistants that Learn from Demonstration and Adapt through Mixed-Initiative Dialogue. 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 145, 19 pages. doi:10.1145/3746059.3747700

[4] Riku Arakawa, Hiromu Yakura, and Mayank Goel. 2024. PrISM-Observer: Intervention Agent to Help Users Perform Everyday Procedures Sensed using a Smartwatch. 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 15, 16 pages. doi:10.1145/3654777.3676350

[5] Riku Arakawa, Hiromu Yakura, Vimal Mollyn, Suzanne Nie, Emma Russell, Dustin P. DeMeo, Haarika A. Reddy, Alexander K. Maytin, Bryan T. Carroll, Jill Fain Lehman, and Mayank Goel. 2023. PrISM-Tracker: A Framework for Multimodal Procedure Tracking Using Wearable Sensors and State Transition Information with User-Driven Handling of Errors and Uncertainty. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 156 (Jan. 2023), 27 pages. doi:10.1145/3569504

[6] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R. Millán, and Daniel Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042. doi:10.1016/j.patrec.2012.12.014 Smart Approaches for Human Action Recognition.

[7] Wenqiang Chen, Shupei Lin, Elizabeth Thompson, and John Stankovic. 2021. SenseCollect: We Need Efficient Ways to Collect On-body Sensor-based Human Activity Data! *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 91 (Sept. 2021), 27 pages. doi:10.1145/3478119

[8] Hai Dang, Frederik Brudy, George Fitzmaurice, and Fraser Anderson. 2023. WorldSmith: Iterative and Expressive Prompting for World Building with a Generative AI. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 63, 17 pages. doi:10.1145/3586183.3606772

[9] 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. 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 600, 22 pages. doi:10.1145/3613904.3642579

[10] Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, et al. 2025. Embodied AI Agents: Modeling the world. *arXiv preprint arXiv:2506.22355* (2025).

[11] Eric Horvitz. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, 159–166. doi:10.1145/302979.303030

[12] Xiyun Hu, Chenfei Zhu, Shao-Kang Hsia, Dizhi Ma, Rahul Jain, and Karthik Ramani. 2026. ARify: Leveraging Narrated Instructional Videos to Create Augmented Reality Tutorials for Procedural Tasks. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*. Association for Computing Machinery, New York, NY,

USA, Article 887, 23 pages. doi:10.1145/3772318.3790715

[13] David Jansson and Steven Smith. 1991. Design fixation. *Design Studies* 12 (01 1991). doi:10.1016/0142-694X(91)90003-F

[14] Yanna Jiang, Delong Li, Haiyu Deng, Baihe Ma, Xu Wang, Qin Wang, and Guangsheng Yu. 2026. SoK: Agentic Skills–Beyond Tool Use in LLM Agents. *arXiv preprint arXiv:2602.20867* (2026).

[15] Cholmin Kang, Hyunwoo Jung, and Youngki Lee. 2019. Towards Machine Learning with Zero Real-World Data. In *The 5th ACM Workshop on Wearable Systems and Applications* (Seoul, Republic of Korea) *(WearSys '19)*. Association for Computing Machinery, New York, NY, USA, 41–46. doi:10.1145/3325424.3329662

[16] David Kirsh and Paul Maglio. 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science* 18, 4 (1994), 513–549. doi:10.1207/s15516709cog1804_1

[17] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474* (2017).

[18] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D. Abowd, Nicholas D. Lane, and Thomas Plötz. 2020. IMUTube: Automatic Extraction of Virtual on-body Accelerometry from Video for Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 3, Article 87 (Sept. 2020), 29 pages. doi:10.1145/3411841

[19] Hyeokhyen Kwon, Bingyao Wang, Gregory D. Abowd, and Thomas Plötz. 2021. Approaching the Real-World: Supporting Activity Recognition Training with Virtual IMU Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 111 (sep 2021), 32 pages. doi:10.1145/3478096

[20] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data Augmentation for Time Series Classification using Convolutional Neural Networks. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*. Riva Del Garda, Italy. https://shs.hal.science/halshs-01357973

[21] 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. ImaginateAR: AI-Assisted In-Situ Authoring in Augmented Reality. 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 52, 21 pages. doi:10.1145/3746059.3747635

[22] Zikang Leng, Amitrajit Bhattacharjee, Hrudhai Rajasekhar, Lizhe Zhang, Elizabeth Bruda, Hyeokhyen Kwon, and Thomas Plötz. 2024. IMUGPT 2.0: Language-Based Cross Modality Transfer for Sensor-Based Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3, Article 112 (Sept. 2024), 32 pages. doi:10.1145/3678545

[23] Zikang Leng, Megha Thukral, Yaqi Liu, Hrudhai Rajasekhar, Shruthi K Hiremath, Jiaman He, and Thomas Plötz. 2026. Agentsense: Virtual sensor data generation using llm agents in simulated home environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 1891–1899.

[24] David Li, Nels Numan, Xun Qian, Yanhe Chen, Zhongyi Zhou, Evgenii Alekseev, Geonsun Lee, Alex Cooper, Min Xia, Scott Chung, Jeremy Nelson, Xiuxiu Yuan, Jolica Dias, Tim Bettridge, Benjamin Hersh, Michelle Huynh, Konrad Piascik, Ricardo Cabello, David Kim, and Ruofei Du. 2025. XR Blocks: Accelerating Human-centered AI + XR Innovation. arXiv:2509.25504 [cs.HC] https://arxiv.org/abs/2509.25504

[25] Jiyang Li, Lin Huang, Siddharth Shah, Sean J Jones, Yincheng Jin, Dingran Wang, Adam Russell, Seokmin Choi, Yang Gao, Junsong Yuan, et al. 2023. Signring: Continuous american sign language recognition using imu rings and virtual imu data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 3 (2023), 1–29.

[26] Xi'ang Li, Jinqi Luo, and Rabih Younes. 2020. ActivityGAN: generative adversarial networks for data augmentation in sensor-based human activity recognition. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers* (Virtual Event, Mexico) *(UbiComp/ISWC '20 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 249–254. doi:10.1145/3410530.3414367

[27] Yunzhi Li, Vimal Mollyn, Kuang Yuan, and Patrick Carrington. 2024. WheelPoser: Sparse-IMU Based Body Pose Estimation for Wheelchair Users. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (St. John's, NL, Canada) *(ASSETS '24)*. Association for Computing Machinery, New York, NY, USA, Article 8, 17 pages. doi:10.1145/3663548.3675638

[28] Fengzhou Liang, Tian Min, Chengshuo Xia, and Yuta Sugiura. 2025. Vsens Toolkit: AR-based Open-ended System for Virtual Sensors. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 754, 5 pages. doi:10.1145/3706599.3721177

[29] Fengzhou Liang, Tian Min, Chengshuo Xia, and Yuta Sugiura. 2026. Vsens: Incorporating XR into the Process of Collecting Virtual IMU Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 10, 1, Article 10 (March 2026), 31 pages. doi:10.1145/3790110

[30] Di Liu, Jingwen Bai, Zhuoyi Zhang, Yilin Zhang, Zhenhao Zhang, Jian Zhao, and Pengcheng An. 2025. TherAIssist: Assisting Art Therapy Homework and Client-Practitioner Collaboration through Human-AI Interaction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 3, Article 113 (Sept. 2025), 38 pages. doi:10.1145/3749497

[31] Di Liu, Jingwen Bai, Zhuoyi Zhang, Yilin Zhang, Zhenhao Zhang, Jian Zhao, and Pengcheng An. 2025. TherAIssist: Assisting Art Therapy Homework and Client-Practitioner Collaboration through Human-AI Interaction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 3, Article 113 (Sept. 2025), 38 pages. doi:10.1145/3749497

[32] Jiacheng Liu, Xiaohan Zhao, Xinyi Shang, and Zhiqiang Shen. 2026. Dive into Claude Code: The Design Space of Today's and Future AI Agent Systems. arXiv:2604.14228 [cs.SE] https://arxiv.org/abs/2604.14228

[33] Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2024. DirectGPT: A Direct Manipulation Interface to Interact with Large Language Models. 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 975, 16 pages. doi:10.1145/3613904.3642462

[34] Muhammad Naeem, Wilson Ozuem, Kerry Howell, and Silvia Ranfagni. 2023. A Step-by-Step Process of Thematic Analysis to Develop a Conceptual Model in Qualitative Research. *International Journal of Qualitative Methods* 22 (2023), 16094069231205789. arXiv:https://doi.org/10.1177/16094069231205789 doi:10.1177/16094069231205789

[35] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. 2000. A model for types and levels of human interaction with automation. *Trans. Sys. Man Cyber. Part A* 30, 3 (May 2000), 286–297. doi:10.1109/3468.844354

[36] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 2, 22 pages. doi:10.1145/3586183.3606763

[37] Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. 2025. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv preprint arXiv:2501.01428* (2025).

[38] Vitor Fortes Rey, Peter Hevesi, Onorina Kovalenko, and Paul Lukowicz. 2019. Let there be IMU data: generating training data for wearable, motion sensor based activity recognition from monocular RGB videos. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers* (London, United Kingdom) *(UbiComp/ISWC '19 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 699–708. doi:10.1145/3341162.3345590

[39] Donald A. Schön. 1983. *The Reflective Practitioner: How Professionals Think in Action.* Temple Smith, London.

[40] Jingyu Shi, Rahul Jain, Seunggeun Chi, Hyungjun Doh, Hyung-gun Chi, Alexander J. Quinn, and Karthik Ramani. 2025. CARING-AI: Towards Authoring Context-aware Augmented Reality INstruction through Generative Artificial Intelligence. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 29, 23 pages. doi:10.1145/3706598.3713348

[41] Lucy A. Suchman. 1987. *Plans and Situated Actions: The Problem of Human-Machine Communication.* Cambridge University Press, Cambridge, England.

[42] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation. 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 644, 26 pages. doi:10.1145/3613904.3642400

[43] Tencent Hunyuan 3D Digital Human Team. 2025. HY-Motion 1.0: Scaling Flow Matching Models for Text-To-Motion Generation. *arXiv preprint arXiv:2512.23464* (2025).

[44] Samangi Wadinambiarachchi, Ryan M. Kelly, Saumya Pareek, Qiushi Zhou, and Eduardo Velloso. 2024. The Effects of Generative AI on Design Fixation and Divergent Thinking. 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 380, 18 pages. doi:10.1145/3613904.3642919

[45] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. 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 385, 22 pages. doi:10.1145/3491102.3517582

[46] Chengshuo Xia, Xinyi Chen, Lingtao Huang, Tian Min, and Daxing Zhang. 2025. Open-Source Virtual IMU Sensor Platform for Developing a Customized Wearable Human Activity Recognition System. *IEEE Transactions on Instrumentation and Measurement* 74 (2025), 1–14. doi:10.1109/TIM.2025.3548063

[47] Chengshuo Xia, Xinrui Fang, Riku Arakawa, and Yuta Sugiura. 2022. VoLearn: A Cross-Modal Operable Motion-Learning System Combined with Virtual Avatar and Auditory Feedback. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 81 (July 2022), 26 pages. doi:10.1145/3534576

[48] Chengshuo Xia, Qingyuan Peng, Zeyuan Fan, Tian Min, Daxing Zhang, and Congsi Wang. 2025. vCapTouch: Interactive Touch Sensing Data Synthesis for Hand Gesture Recognition Based on Digital Twin. *IEEE Internet of Things Journal* 12,

13 (2025), 23823–23834. doi:10.1109/JIOT.2025.3553561

[49] Chengshuo Xia and Yuta Sugiura. 2022. Virtual IMU Data Augmentation by Spring-Joint Model for Motion Exercises Recognition without Using Real Data. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers* (Cambridge, United Kingdom) *(ISWC '22)*. Association for Computing Machinery, New York, NY, USA, 79–83. doi:10.1145/3544794.3558460

[50] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. 2018. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE.

[51] Renjun Xu and Yang Yan. 2026. Agent skills for large language models: Architecture, acquisition, security, and the path forward. *arXiv preprint arXiv:2602.12430* (2026).

[52] Ziyi Xuan, Yiwen Wu, Xuhai Xu, Vinod Namboodiri, Mooi Choo Chuah, and Yu Yang. 2025. Design and Evaluation of Generative Agent-based Platform for Human-Assistant Interaction Research: A Tale of 10 User Studies. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 4, Article 229 (Dec. 2025), 46 pages. doi:10.1145/3770661

[53] A. D. Young, M. J. Ling, and D. K. Arvind. 2011. IMUSim: A simulation environment for inertial sensing algorithm design and evaluation. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE, 199–210.

[54] Qianwen Zhong, Wenhui Liu, Hongping Zhou, Ping Lu, and Wenzhe Shi. 2026. 3D Gaussian Splatting for Scene Editing and Completion. In *Proceedings of the 2nd International Conference on Artificial Intelligence, Digital Media Technology and Social Computing (ICAIDS '26)*. Association for Computing Machinery, New York, NY, USA, 14–25. doi:10.1145/3806262.3806264