A Bit of Clarification: Probabilistic User-in-the-Loop Disambiguation of Multimodal Input in AR

Joseph Rupertus

Advisor: Parastoo Abtahi

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Engineering

Department of Computer Science

Princeton University

May 2026

I hereby declare that I am the sole author of this thesis.

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Joseph Rupertus

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Joseph Rupertus

Abstract

Multimodal inputs in Augmented Reality (AR) are inherently imprecise, noisy, and underspecified, meaning the same input can map to multiple valid interpretations. This is not a limitation of recognition or inference systems, but rather it is fundamentally an interaction problem. We introduce a proof-of-concept AR system that addresses this through user-in-the-loop disambiguation, using a Dynamic Bayesian Network to fuse gaze, voice, and gesture inputs into probability distributions over target and action spaces. The system enters disambiguation mode when no candidate is sufficiently confident, presenting audio and visual feedback to guide users toward a single clarifying input. We evaluate the system through a two-part user study: a multimodal elicitation study and a disambiguation mode comparison. Our fusion system correctly predicted intent in 53.1% of free-form elicitation trials, with the correct intent appearing in the candidate set in 83.3% of trials. Participants successfully disambiguated after a single input in 82.8% of trials.

Acknowledgements

Thank you to my advisor, Parastoo Abtahi, for her close collaboration and guidance throughout the many twists and turns of this research process. Thank you to my second reader, Anirudha Majumdar, for his support of this project and for the robotics course that introduced me to HRI. Thank you also to Andrés Monroy-Hernández, whose HCI course first sparked my interest in this field and shaped the direction of my research. Thank you to my lab members, especially Lauren Wang, whose expertise was invaluable to this work. Thank you also to Mo Kari, Cyrus Vachha, Ching-Yi Tsai, Varun Rao, and Michael Huang for their help and advice. Finally, thank you to my parents for their love and support, and all of my friends here at Princeton for everything else. This would not be possible without all of you.

Contents

Abstract iii

Acknowledgements iv

1 Introduction **1**

2 Related Work **5**

2.1 Multimodal Fusion and Intent Inference 5

2.1.1 Probabilistic Models for Intent Prediction 6

2.2 User-in-the-Loop Disambiguation 7

2.2.1 Implicit (One Step) Disambiguation 7

2.2.2 Explicit (Two Step) Disambiguation 8

3 Gesture Exploration **9**

3.1 Motivation 9

3.1.1 Gesture Customization and Personalization 10

3.1.2 Zero-Shot Gesture Understanding 10

3.2 Approaches 11

3.2.1 VLM Understanding 11

3.2.2 Rule-based Description + LLM Understanding 12

3.2.3 Visual Hand Rendering + VLM Understanding 13

3.3 Takeaways 14

3.4 Heuristics Implementation 15

4 Multimodal Fusion System **16**

4.1 Dynamic Bayesian Network 17

4.2 Transition Model 19

4.3 Observation Model 20

4.3.1 Gaze Input 20

4.3.2 Voice Input 21

4.3.3 Gesture Input 22

4.4 Intent Inference 24

4.4.1 Intent Resolution 25

4.4.2 Intent Disambiguation 25

5 Disambiguation Design **27**

5.1 One-Bit Disambiguation 28

5.2 Location and Identity 28

5.2.1 Give Targets More Identity 29

5.2.2 Give Actions Spatial Location 29

5.3 Target Disambiguation Design 30

5.4 Action Disambiguation Design 32

6 User Study **34**

6.1 Phase 1: Multimodal Elicitation 34

6.1.1 Elicitation Tasks 35

6.1.2 Modality Constraints 36

6.1.3 System Validation 38

6.2 Phase 2: Disambiguation Comparison 38

6.2.1 Target Selection Comparison 39

6.2.2 Action Selection Comparison 40

6.2.3 Feedback Collection 40

7 Evaluation **41**

7.1 Elicitation Results 41

7.2 Fusion Validation 43

7.3 Disambiguation Comparison 45

7.3.1 Target Disambiguation 45

7.3.2 Action Disambiguation 47

7.4 Design Takeaways 49

8 Applications **51**

8.1 Embedded Agents 52

8.2 Embodied Agents 53

8.3 Disembodied Agents 54

9 Conclusion **56**

Chapter 1

Introduction

Augmented Reality (AR) systems interpret user intent through multiple input modalities, including gaze direction, voice, and mid-air hand gestures, to enable interactions with menus, virtual objects, and smart environment controls. Making sense of these inputs together is a core challenge in AR interaction design. For instance, when a user is interacting with a smart lamp to turn it on, they may say *"turn on the lamp,"* or look at the lamp and say *"turn on,"* or look at the lamp and perform a pinch gesture. Any of these multimodal input strategies should result in the same action output: the lamp turns on.

In many cases like this, a single input is enough to confidently determine the user's intent. But in general, predicting intent via arbitrary voice inputs or gestures is inherently ambiguous. For instance, consider the case where there are *two* lamps in the user's field of view, and they say *"turn on the lamp."* How can the system know which lamp the user meant to turn on? Or consider the case where there is a lamp that has the ability to change both brightness and color, and the user looks at the lamp and performs a swipe gesture upward. This gesture is indicative of an increase

of some kind, but how can the system know whether the user meant to increase its brightness or increase its hue?

Clearly, it is not feasible to confidently map any arbitrary multimodal input to a specific intent, because the same input can map to multiple valid intents. Voice and gesture are inherently ambiguous, and gaze is inherently noisy and imprecise. This is not a problem that can be solved by more powerful algorithms or vision models. Even a theoretically perfect perception system couldn't resolve the ambiguity, because the ambiguity is in the user's underspecified input, not in the system's ability to perceive it. Instead, this presents an interaction problem: how to communicate the system's uncertainty to the user and allow them to resolve it with minimal additional effort. This work explores user-in-the-loop disambiguation strategies, where real-time uncertainty feedback guides users to provide targeted clarification until the system can confidently predict their intent.

We present a proof-of-concept Meta Quest application in which users interact with objects using these strategies. Our system follows the foundational architecture of XWand (Wilson and Shafer, 2003), which established the use of a Bayesian Network to fuse pointing and speech inputs for smart home control. We extend this work by bringing it into AR and using a Dynamic Bayesian Network to fuse hand gesture, gaze, and voice modalities (Han et al., 2025). Our system maintains beliefs over the intended referent and action, updating with each new input. When the distribution is insufficiently peaked to resolve to a single intent, the system enters disambiguation mode, prompting the user for additional input.

We created four audio and visual feedback modes each for target and action disambiguation, which communicate uncertainty to the user and encourage them to clarify. We motivate this through the idea of *one-bit disambiguation*, in which a single clarifying input is sufficient to resolve ambiguity. We designed target and ac-

tion disambiguation distinctly, based on the observation that targets have intrinsic location but ambiguous identity, while actions have distinct identity but no intrinsic location. Our disambiguation designs compensate for what each leaves unspecified: the target disambiguation modes involve surfacing additional identifiers to encourage disambiguating through them, and action disambiguation modes involve various ways of embedding candidate actions as spatial targets in the scene.

We evaluated our system through a two-part user study. The first phase was an elicitation study in which we aimed to understand what types of multimodal inputs users would naturally perform to accomplish a given task. Participants mostly preferred voice while acknowledging the benefits of gestures, and half performed both voice and gesture simultaneously when allowed. We ran this phase using a Wizard-of-Oz setup and later evaluated our fusion system offline using the logged data. The second phase was a comparison of our disambiguation modes, where we collected feedback and usability data for each condition as part of our design space exploration. Participants mostly preferred the audio feedback mode for target disambiguation and the visual button menu for action disambiguation.

The contributions of this thesis are:

- A multimodal fusion system using a Dynamic Bayesian Network to maintain probability distributions over target and action spaces in AR
- A set of eight disambiguation mode designs (four for target disambiguation and four for action disambiguation) involving audio and visual elements that communicate uncertainty and guide user clarification
- A two-part user study evaluating natural multimodal input behavior and comparing disambiguation modes across usability dimensions

- A framework for applying this system across three categories of real-world agents: embedded, embodied, and disembodied

The remainder of this thesis begins with a review of related work, followed by our explorations in free-form gesture recognition and the insights that informed our final design approach. We then describe the implementation of our multimodal fusion system and the design of our user-in-the-loop disambiguation strategies. We detail our user study procedure and discuss the results and key findings. We then explore applications of our system across domains, from IoT to robotic control. Finally, we discuss limitations and directions for future work.

Chapter 2

Related Work

2.1 Multimodal Fusion and Intent Inference

A core challenge in multimodal systems is fusing inputs across different modalities such as gaze, speech, and gestures to infer user intent. This problem was pioneered by the "Put-That-There" system (Bolt, 1980), where pointing gestures and speech commands were combined so that each could fill the gaps of the other. Subsequent work builds on this: Oviatt (1999) demonstrates that multimodal input can reduce error in intent prediction, especially when speech is augmented with gestures. Kaiser et al. (2003) creates a probabilistic architecture to formalize how to perform this fusion using ranked n-best lists with associated probabilities.

Gesture and speech have a natural pairing for object–action resolution. Piumsomboon et al. (2014) finds that speech combined with gestures outperforms direct manipulation alone, and elicitation studies such as Williams et al. (2020) and Zhou et al. (2022) confirm that users naturally gravitate toward gesture for spatial reference and speech for action specification. In AR specifically, Chopra and Maurer (2020) finds

that users prefer multimodal interaction for smart environment control over a single modality. GazePointAR (Lee et al., 2024) incorporates a third modality, eye gaze, to disambiguate uncertain pronoun references like "this" or "that" in speech. Wang et al. (2021) finds that integrating all three of these modalities is superior in efficiency to gesture and speech alone.

More recent work extends beyond fixed input grammars into free-form understanding. GesPrompt (Hu et al., 2025) uses LLMs to identify unspecified parameters in speech and resolve them through gesture, supporting a richer range of gesture types including shape, size, and rotation. In addition, recent work such as GestureGPT (Zeng et al., 2024) and Gestura (Li et al., 2025) use LLMs and LVLMs to recognize a broader, open gesture space not limited to predefined categories. However, even when inputs are correctly recognized, intent can still be genuinely unclear, and the same gesture or speech command may map to multiple valid interpretations. This is the problem of intent disambiguation that our work addresses.

2.1.1 Probabilistic Models for Intent Prediction

Because multimodal inputs are underspecified, intent must be inferred probabilistically from uncertain evidence. XWand (Wilson and Shafer, 2003) establishes the foundational approach of a Bayes Network that fuses speech with pointing via a wand to determine the intended action and command in a smart home setting. Han et al. (2025) extends this with a Dynamic Bayesian Network (DBN) that continuously updates beliefs as new observations arrive, adapting to varying environments and integrating contextual world knowledge through LLM elicitation of prior probabilities.

Many other probabilistic model types have been applied to this problem: Jain and Argall (2019) applies recursive Bayes filtering to infer human intent from action observations in robotics, and He et al. (2026) formulates intent inference for robots

as a POMDP. Chai et al. (2004) addresses the problem using graph matching over probabilistic representations, while Velloso and Morimoto (2021) uses entropy over a candidate set as a measure of uncertainty.

Schwarz et al. (2011) applies probabilistic input handling to user interfaces, introducing a framework that maintains a probability distribution of intents using Monte Carlo sampling, rather than immediately committing to one interpretation of the input. Schwarz et al. (2015) extends this work by adding visual feedback of uncertainty through the UI, displaying multiple alternatives in a single interface. Our work builds upon these principles, using a DBN to maintain a probability distribution over each referent and each action, and providing real-time uncertainty feedback visually to the user to disambiguate their intent.

2.2 User-in-the-Loop Disambiguation

2.2.1 Implicit (One Step) Disambiguation

We first describe *implicit* disambiguation strategies, where the system communicates uncertainty to the user in some form, but resolves it using natural, passive interactions they are already performing. In other words, it does not require them to perform any deliberate extra steps. Object pointing snaps selection to discrete objects rather than using a continuous cursor position, reducing ambiguity about which object the user intended to select (Guiard et al., 2004). Some systems work by correlating user input with target motion, allowing users to select targets by matching their input motion to targets (Velloso and Morimoto, 2021; Clarke and Gellersen, 2020). Others act on the input signal itself, steering users' gaze toward the intended target by visually outlining the predicted candidate (Sidenmark et al., 2020; Çığ Karaman and Sezgin, 2018). Still others act on the display, artificially increasing target size or reducing target

distance to facilitate precise selection (Balakrishnan, 2004). Guillon et al. (2015) establish key principles for how candidates should be surfaced to the user during target selection. Tsai et al. (2026) demonstrates how feedforward visualizations can be adapted for target selection in AR, characterizing pointer archetypes and visual signifiers for communicating uncertainty.

2.2.2 Explicit (Two Step) Disambiguation

There are also *explicit* disambiguation strategies, in which the system communicates uncertainty and requires the user to take an additional clarification step to resolve it, often via a different modality than the original input. For instance, Chatterjee et al. (2015) uses gaze as a coarse input, followed by a gesture to confirm the precise selection. Some systems combine eye gaze and head movement to disambiguate (Kytö et al., 2018), and others combine freehand manipulation with eye gaze as disambiguation (Chen et al., 2020). Sidenmark et al. (2022) switches to a fallback modality in cases of eye tracking uncertainty, which the user then uses to point accurately. Alternatively, Wu et al. (2015) narrows to a candidate set in a first step, then uses a second step of 1D target selection to precisely select within it. Some systems involve an even more explicit disambiguation step, asking the user directly through a UI in cases of uncertainty (Lin et al., 2023), or verbally when a robot is uncertain of what it should do (Ren et al., 2023). Our work extends this paradigm with a user-in-the-loop disambiguation interaction specifically for AR, incorporating both target and action disambiguation, where candidates are surfaced spatially and multiple input modalities are available for resolution.

Chapter 3

Gesture Exploration

Throughout the research process, we explored various methods for recognizing hand gestures and mapping them to user intents, specifically targeting an open gesture vocabulary without predefined classifications. For our final system, we decided to use a limited gesture set designed around heuristics to ensure reliability and robustness for the user, allowing us to focus research effort on the multimodal fusion architecture and disambiguation design, the core contribution of this thesis. In this chapter, we describe the different methods we originally attempted to integrate open gesture understanding into our system. The challenges we encountered directly informed our understanding of what makes gesture recognition difficult in this domain, and shaped the heuristic design we ultimately adopted. We leave full integration of open gesture recognition into a multimodal system as future work.

3.1 Motivation

Mid-air hand gestures are a crucial interaction method for AR applications. Researchers have established standardized frameworks for understanding and describing

gestural input. Hosseini et al. (2023) proposes a set of 22 general-purpose gestures providing a common vocabulary users can learn once and apply broadly, while Choi et al. (2014) develops a comprehensive gesture taxonomy capable of encoding any gesture, including complex dynamic and bimanual movements. However, both approaches still require explicit gesture-to-action mappings, placing the burden of design on system designers or end users. Creating these vocabularies is itself difficult and error-prone: Xia et al. (2022) identifies 13 critical factors a good gesture vocabulary must satisfy, and they find that expert-led, user-led, and computationally based design approaches each have distinct shortcomings.

3.1.1 Gesture Customization and Personalization

Prior research has introduced systems that allow users to customize gestures, delegating the gesture-to-action mapping directly to the user. Wobbrock et al. (2009) elicits gestures from non-technical users by presenting the effect of a command and asking users to perform its cause, finding that designer-created gestures often fail to reflect actual user behavior. Shahi et al. (2024) extends this to a one-shot setting, introducing a system that can recognize a new gesture after just a single demonstration by the user, circumventing the need for extensive training sets with meta-learning techniques. While personalization allows for more flexibility, these systems still require explicit definition: users must consciously decide what gesture maps to what action, presupposing they know in advance what gestures they want to use.

3.1.2 Zero-Shot Gesture Understanding

GestureGPT (Zeng et al., 2024) represents a paradigm shift away from fixed-vocabulary approaches, such as Hidden Markov Models (Tanguay, 1995), and toward zero-shot free-form understanding, using LLM agents to interpret arbitrary hand gestures with-

out predefined mappings or user demonstrations. However, its multi-turn agent dialogues produce an average latency of 227 seconds, rendering it impractical for real-time interaction, and its reliance on text-only reasoning limits its understanding of fine-grained hand dynamics. Gestura (Li et al., 2025) addresses these shortcomings with a Large Vision Language Model (LVLM) backbone augmented by a Landmark Processing Module, which extracts hand keypoints and augments the video feed with them to provide both visual and anatomical context. Gestura achieves 1.6 second latency, which is substantially more practical, though still far from the real-time performance required for interactive AR. The problem of real-time free-form gesture understanding remains largely unsolved.

3.2 Approaches

3.2.1 VLM Understanding

At the start, we tried passing raw egocentric video directly to a Vision Language Model (VLM) to understand intent. We tested both Gemini 2.5 Flash and Gemini 3 Pro to compare the feasibility of the task with both lightweight and heavyweight models. We first prompted the model with a raw video file, and instructed it to recognize the gesture performed in the video and output a natural language description of the gesture in relation to the physical environment. Then, we separately prompted the model with the same natural language description, along with a JSON file containing the interactive objects in the environment and their associated actions, and instructed it to predict the most likely user intent.

This approach proved to be very difficult. For both Gemini 2.5 Flash and Gemini 3 Pro, the model could not recognize nuanced hand gestures, especially with quick motions. It performed well at recognizing static hand poses, but failed to generalize

over a video stream in order to accurately classify a full motion. It would often misclassify, for example, a pinch gesture as a pointing gesture, because it could not correctly recognize the quick pinch motion. It also frequently reversed the direction of motion (i.e. it would believe an upward swipe to be a downward swipe), likely due to an incorrect understanding of the frame sequence, despite explicit prompting. Furthermore, it often hallucinated the hand physically interacting with objects in the environment, when in reality, it was mid-air and performing a gesture meant to invoke an action through that very recognition. Latency for Gemini 2.5 Flash averaged 5–10 seconds per gesture, far too slow for real-time interaction. Although we expect these models to improve in the future, VLMs alone in their current form cannot accurately classify dynamic gestures.

3.2.2 Rule-based Description + LLM Understanding

Inspired by GestureGPT (Zeng et al., 2024), we created a rule-based pipeline that converted raw egocentric video into a semantic gesture description, then passed it to an LLM for intent prediction. Each frame was processed by MediaPipe to extract 21 hand keypoints, from which we computed geometric features: per-finger flexion values, inter-finger proximity and contact distances, thumb pointing direction, and palm orientation. These continuous features were discretized into categorical states (e.g., extended, bent, closed) and grouped by shared configuration to produce a compact natural language description of the gesture. Subsequent frames were encoded incrementally, recording only features that changed from the previous state, yielding a structured temporal description of the full motion.

This description was then passed to Gemini 2.5 Flash along with a JSON list of interactive objects in the environment, each with their current state and available actions. We prompted the model to reason through the gesture and output a target–

action pair from the object list, or NULL if no meaningful intent could be inferred. While this approach produced more reliable gesture descriptions than raw VLM understanding and reduced latency to approximately 3–5 seconds with Gemini 2.5 Flash, it remained brittle: the rule-based encoding lost spatial and temporal nuance, and intent prediction accuracy was insufficient for practical use.

3.2.3 Visual Hand Rendering + VLM Understanding

Our next approach moved back into the realm of vision models, but specifically chose to highlight and isolate the hand's physical form from the background environment to reduce hallucination. To do this, we extracted the keypoints of the hand using the Meta Hand Tracking SDK, projected them onto a 2D rendering based on their physical 3D world locations, and connected them visually with lines. This created a hand rendering that VLMs are capable of recognizing. We created one rendering per input frame and submitted them to a VLM, and instructed it to output a natural language description of the gesture, and infer the user's intent based on the available actions. This was prone to the same issue as the raw video, where VLMs were unable to recognize nuances in the gesture motion.

To remedy this, we tried a different approach of overlaying select frames on top of each other in a single image, in order to convey both the hand pose and the dynamic motion. This was even less clear to the VLM. We tried using only two poses (the start and stop positions) along with a trajectory line connecting them. This was also unclear. We tried reducing this to only a single pose (the end pose) with a trajectory line showing where the hand had moved. This technique was able to identify the correct motion direction and the correct pose, but still did not provide enough nuance for complex gestures, especially ones where the pose itself changes throughout the motion. The VLM would also sometimes hallucinate the direction of

motion, just as with the raw video, even when the direction was made explicitly clear via arrow. Passing all frames to Gemini 2.5 Flash produced latency of 5–10 seconds, while reducing to a single overlaid frame brought it to approximately 3–5 seconds, which is still insufficient for real-time use. Overall, the recognition was not robust enough, and we concluded that VLMs in their current form are not ready for complex gesture understanding.

3.3 Takeaways

We determined through these experiments that current general-purpose VLMs and LLMs cannot accurately classify dynamic gestures for real-time applications. Even the current state-of-the-art, Gestura (Li et al., 2025), which achieves 1.6 second latency by combining structured hand landmark features with an LVLM backbone, remains too slow for the interactive AR setting we targeted. We identified that we could achieve accurate classifications of both individual static poses and dynamic motion trajectories in all of the methods we attempted, but could never achieve high enough accuracy when both of these elements were combined, i.e. a dynamic motion that changes pose over time. Current VLMs are simply limited in temporal understanding.

To provide gesture inputs to our multimodal fusion system, we decided to implement a strict vocabulary of predefined gesture heuristics, which use rule-based detection to classify specific motions, and fall back to sending a single frame pose to a VLM to recognize symbolic gestures, as this avoids the problems we described about temporal reasoning in VLMs. This simplifies the input space dramatically and provides quick recognition of gestures, allowing us to focus on our core contributions of multimodal fusion (Chapter 4) and disambiguation (Chapter 5). We also determined that the chosen heuristics more accurately fit into our smart home study setup (Chapter 6), and that most plausible gesture interactions with the available actions in the setup

fit these heuristics well. We evaluate this claim more clearly in the discussion (Chapter 7). The heuristics and their implementations are described below.

3.4 Heuristics Implementation

The gesture vocabulary was largely derived through informal testing during development of the multimodal fusion system. We observed that swipe, pinch, and point gestures arose consistently, and these gestures mapped naturally onto the specific actions that are afforded by the smart home objects in our study scene. This informal elicitation process is consistent with prior findings that gesture vocabularies should emerge from natural user behavior (Wobbrock et al., 2009). Accordingly, we implemented the gesture recognition system using the following heuristics:

Swipe. A swipe is detected by accumulating hand displacement across frames. The axis with the greatest total displacement determines the swipe direction, provided the motion exceeds a minimum distance and follows approximately a straight line. The output is a discrete direction label (e.g., left, right, up, down).

Pinch. A pinch gesture is detected when the hand is near the center of the gaze field in a pinching pose. The output is a boolean indicating that a pinch was performed.

Point. A pointing gesture is detected when the hand is near the center of the gaze field and has an extended index finger. The output is a 3D ray along the index finger into the scene, along with a coarse semantic direction label (e.g., left, right, up, down) derived from the same ray direction.

Symbolic. If no heuristic gesture is detected, the system renders the hand configuration as a static image and submits it to a VLM, which recognizes the hand pose. This fallback is useful for static poses (e.g. peace sign, thumbs up, etc.).

Chapter 4

Multimodal Fusion System

Consider a user in an environment of interactive objects. Each object *affords* actions[1] that may be performed to modify its state in some way. To trigger an afforded action, the user provides multimodal inputs: they may gaze at objects, speak in natural language, or perform mid-air hand gestures. We present a multimodal fusion system that infers the user's intent and facilitates disambiguation in cases of uncertainty.

To implement the system, we built a Meta Quest 3 application using Unity 6.3 that communicates with a local Python FastAPI server via WebSocket connection and dispatches API calls to IoT devices. The system infers user intent from the multimodal inputs by converting raw input data into likelihood vectors that are fused by a Dynamic Bayesian Network (see Figure 4.1). Once the system reaches a certain confidence threshold in its belief, it triggers the action. Otherwise, it enters *disambiguation mode* and communicates uncertainty to the user visually or audibly by presenting the remaining candidates and encouraging them to clarify their intent.

[1] An *affordance* is an action possibility available in the environment to an individual (Gibson, 1979). In our system, for example, a speaker affords the *toggle* and *raise volume* actions.

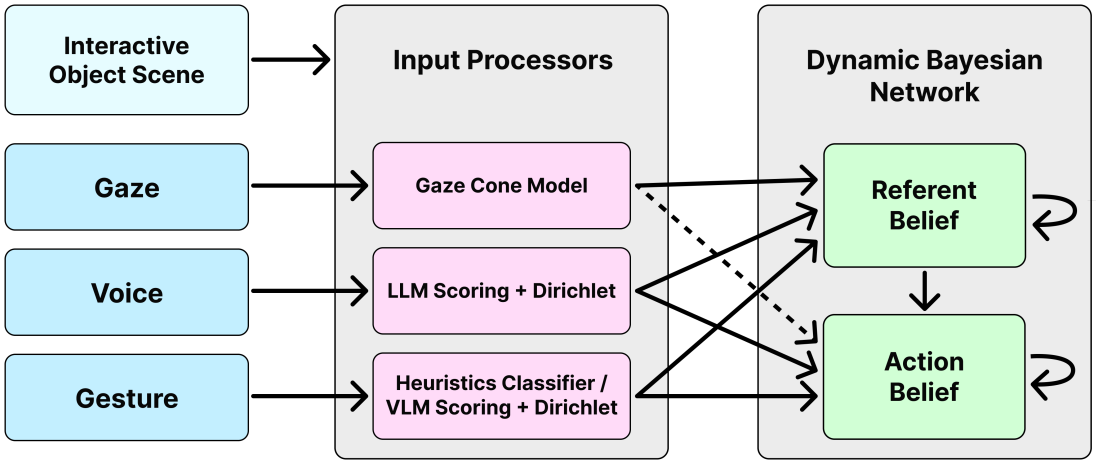


Figure 4.1: Diagram of the fusion system, showing how the referent and action are determined based on multimodal input through the DBN. The dashed line denotes a conditional dependency that applies only in certain operating modes.

We formalize this environment as follows. We represent it as a scene $\mathcal{S} = (\mathcal{R}, \mathcal{A})$, where $\mathcal{R} = \{r_1, r_2, \ldots\}$ is a set of interactive **referents** and $\mathcal{A}$ is the set of all afforded **actions**. Each referent r_i affords a set of actions $\mathcal{A}_i = \{a_{i,1}, a_{i,2}, \ldots\}$, and the full action space is $\mathcal{A} = \bigsqcup_{r_i \in \mathcal{R}} \mathcal{A}_i$, the disjoint union of all actions across the scene. For instance, a scene might contain a lamp affording $\{\textit{brightness up}, \textit{toggle}\}$ and a speaker affording $\{\textit{volume up}, \textit{volume down}\}$. The user intent is defined by a **target–action pair** (r, a): the referent object the user meant to interact with, and the afforded action they meant to trigger. For instance, $(\textit{lamp}, \textit{brightness up})$, $(\textit{speaker}, \textit{toggle})$, and $(\textit{robot}, \textit{move forward})$ are target–action pairs.

4.1 Dynamic Bayesian Network

We use a Dynamic Bayesian Network (DBN) to fuse multimodal inputs and probabilistically model user intent. The DBN provides a robust structure for combining

evidence across modalities, propagating belief over time, and reasoning transparently about uncertainty, goals that are difficult to achieve with purely heuristic or rule-based approaches.

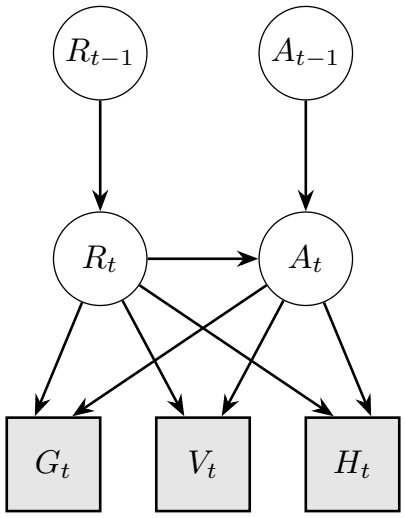


Figure 4.2: The Dynamic Bayesian Network structure.

At each timestep t, the network is defined by two hidden states: the referent $R_t \in \mathcal{R}$ and the action $A_t \in \mathcal{A}$. The system maintains a belief $\text{bel}(R_t)$ over the objects in $\mathcal{R}$, and a belief $\text{bel}(A_t)$ over the actions in $\mathcal{A}$. The network contains one observation node per input modality: gaze (G_t), voice (V_t), and hand gestures (H_t), as described in Figure 4.2. The full network is defined by:[2]

$$P(R_t, A_t, G_t, V_t, H_t \mid R_{t-1}, A_{t-1}) = P(R_t \mid R_{t-1}) \cdot P(A_t \mid A_{t-1}, R_t) \\ \cdot P(G_t \mid R_t, A_t) \cdot P(V_t \mid R_t, A_t) \cdot P(H_t \mid R_t, A_t)$$

[2]Gaze depends only on the referent in standard operation, i.e. $P(G_t \mid R_t, A_t) = P(G_t \mid R_t)$. The dependence on A_t applies only during action disambiguation in the "Spatial Targets" mode, when candidate actions have been assigned physical locations in the scene (see Section 5.4).

4.2 Transition Model

The transition model is used to propagate belief over time. We define the transition model from the previous time slice $t-1$ to the current time slice t as follows:

$$P(R_t = r \mid R_{t-1} = r') \propto \text{Inertia}(r, r'; \lambda_R, |\mathcal{R}|)$$
$$P(A_t = a \mid A_{t-1} = a', R_t = r) \propto \text{Inertia}(a, a'; \lambda_A, |\mathcal{A}|) \cdot \text{Afford}(a, r)$$

The **inertia function** encodes the effect of the previous state, where a weight λ is assigned to self-transition, with the remaining $1-\lambda$ spread uniformly across all other states. This smooths the belief space so that beliefs from prior evidence do not immediately decay out of the model.

$$\text{Inertia}(x, x', \lambda, k) = \begin{cases} \lambda & \text{if } x = x' \\ \dfrac{1-\lambda}{k-1} & \text{otherwise} \end{cases}$$

λ_R is set very high (close to 1) so that referent belief is resistant to change, particularly so that the continuous gaze input does not dominate the other modalities. λ_A is set slightly lower for flexibility in action selection. Both values were tuned empirically.

The **affordance mask** encodes the coupling between referents and actions, assigning 1 to any action a afforded by referent r (by design, each action is assigned to one unique referent), and ϵ to all unafforded pairs. As a result, the action belief is influenced by the referent belief: actions belonging to highly weighted referents receive more weight, while actions inconsistent with the likely referent are suppressed.

$$\text{Afford}(a, r) = \begin{cases} 1 & \text{if } a \in \mathcal{A}_r \\ \epsilon & \text{otherwise} \end{cases}$$

4.3 Observation Model

The system accepts three different input modalities: gaze, voice, and hand gestures, each represented by observation nodes in the DBN. The voice and gesture modalities require a controller button to be pressed and held throughout the input in order to record them. To infer intent, we combine heterogeneous likelihood sources: a gaze cone model, Dirichlet-sampled likelihoods from LLM/VLM scores for voice and symbolic gestures, and heuristic classifiers for swipe, pinch, and pointing gestures, all of which are not necessarily calibrated to the same probability space. We therefore characterize our system as an *approximate* Bayesian fusion framework, following established practice in other multimodal intent inference systems (Han et al., 2025). Robust integration of likelihood sources across modalities remains a direction for future work.

4.3.1 Gaze Input

We approximate gaze input based on the current position and orientation of the headset.[3] Every frame, we process the current headset state and update the gaze observation likelihood in the DBN.

Let $\mathbf{g}_t \in \mathbb{R}^3$ be the unit vector along the headset's forward direction at time t, and let $\mathbf{h}_t \in \mathbb{R}^3$ be the headset position. For each object r at position $\mathbf{p}_r$, we compute the gaze alignment angle:

$$\theta_r = \arccos\left(\mathbf{g}_t \cdot \frac{\mathbf{p}_r - \mathbf{h}_t}{\|\mathbf{p}_r - \mathbf{h}_t\|}\right)$$

The gaze likelihood is modeled as a scaled Gaussian over θ_r, so that objects within

[3]Meta Quest 3 does not support eye tracking.

the user's gaze are boosted accordingly:

$$P(G_t \mid R_t = r) \propto \max\left(\epsilon_R,\ w_R \exp\left(\frac{-\theta_r^2}{2\sigma_{G,R}^2}\right)\right)$$

where $\sigma_{G,R}$ controls the angular spread of the gaze cone, w_R is the weight coefficient, and ϵ_R is a floor value.

In most cases, gaze does not inform the action likelihood, i.e. $P(G_t \mid A_t)$ is uniform over all actions. However, during action disambiguation in the "Spatial Targets" mode (Section 5.4), each candidate action a is assigned a spatial position $\mathbf{p}_a$ in the scene, and the gaze likelihood over actions is computed with its own parameters:

$$P(G_t \mid A_t = a) \propto \max\left(\epsilon_A,\ \exp\left(\frac{-\theta_a^2}{2\sigma_{G,A}^2}\right)\right)$$

where θ_a is the gaze alignment angle to $\mathbf{p}_a$, $\sigma_{G,A}$ is a narrower cone angle, and ϵ_A is set to a high floor to slow belief convergence.

4.3.2 Voice Input

We use the Meta Voice SDK to transcribe voice inputs as text. When voice is detected, we submit the transcript to an LLM, which returns a confidence score $s \in [0, 1]$ for each referent and each action independently, representing how likely it is that the voice input matches that referent or action.[4] For example, given the input *"turn up the brightness,"* the LLM assigns $s_t(\textit{lamp}) = 1$, $s_t(\textit{brightness up}) = 1$, and near-zero scores to unrelated referents and actions.

Since LLM scores are not calibrated probabilities, treating raw scores as Bayesian likelihoods would distort the inference. We instead use the scores as concentration

[4]For this task, we use Gemini 2.5 Flash.

parameters for a Dirichlet distribution that we then sample from, following Han et al. (2025). This approach introduces uncertainty over the likelihood, producing a more robust probability model that reflects the inherent unreliability of raw LLM confidence scores. We sharpen the scores by raising them to exponents $\rho_R, \rho_A \geq 1$ and then scale them by a concentration hyperparameter w:

$$\alpha_r^R = w \cdot s_t(r)^{\rho_R} \qquad \alpha_a^A = w \cdot s_t(a)^{\rho_A}$$

We then sample the Dirichlet distributions to receive a likelihood value:

$$P(V_t \mid R_t) \sim \text{Dirichlet}(\alpha^R) \qquad P(V_t \mid A_t) \sim \text{Dirichlet}(\alpha^A)$$

4.3.3 Gesture Input

We use the Meta Hand Tracking SDK to continuously track the positions of 26 keypoints per hand. At each frame, we process the current keypoints, aiming to *classify* a gesture. We classify gestures using a set of heuristics (swipe, pinch, point, and symbolic), for which we describe the specific implementations in Section 3.4. When a gesture is detected, we update the gesture observation likelihoods as follows.

Swipe. The detected direction label is matched against each action's `swipe_direction` identifier:

$$P(H_t \mid A_t = a) \propto \begin{cases} 1 & \text{if } \texttt{swipe\_direction}(a) = \text{direction} \\ \epsilon & \text{otherwise} \end{cases}$$

Actions are preconfigured with `swipe_direction` identifiers. For example, a *brightness up* action has a `swipe_direction` of `up`, but a discrete action like *toggle* does not have one at all. Referent likelihoods are not affected by swipe gestures.

Pinch. The pinch boolean is matched against actions tagged as `pinchable`:

$$P(H_t \mid A_t = a) \propto \begin{cases} 1 & \text{if } \texttt{pinchable}(a) \\ \epsilon & \text{otherwise} \end{cases}$$

Actions are preconfigured with `pinchable` identifiers. For example, a *toggle* action has a `pinchable` identifier, but a continuous gesture like *brightness up* does not. Referent likelihoods are not affected by pinch gestures.

Point. The pointing ray direction is used to compute a Gaussian likelihood over the angle θ between the ray and the direction to each object or action's world position, in the same way as the gaze model:[5]

$$P(H_t \mid R_t = r) \propto \exp\left(\frac{-\theta_r^2}{2\sigma_{P,R}^2}\right) \qquad P(H_t \mid A_t = a) \propto \exp\left(\frac{-\theta_a^2}{2\sigma_{P,A}^2}\right)$$

where $\sigma_{P,R}$ and $\sigma_{P,A}$ are cone widths.

In some cases, the pointing direction may also be mapped to a semantic direction label $\texttt{sem} \in \{\texttt{left}, \texttt{right}, \texttt{up}, \texttt{down}\}$. This label is used to boost the likelihood of objects and actions whose `position` identifiers match:[6]

$$P(H_t \mid R_t = r) \propto \begin{cases} \beta & \text{if } \texttt{position}(r) = \texttt{sem} \\ 1 & \text{otherwise} \end{cases}$$

where β is a very large boost multiplier. The `position` identifiers are assigned at runtime: relative to the user's forward direction, the leftmost and rightmost objects receive `left` and `right` labels respectively, and similarly for `up` and `down`.

[5]The action likelihood only applies in the "Spatial Targets" mode (Section 5.4).
[6]The action boost only applies in the "Spatial Targets" mode (Section 5.4).

Symbolic. We use a VLM as a fallback to our heuristics in the case that none were detected. The VLM pose recognition produces a confidence score $s \in [0, 1]$ per object and per action.[7] As with voice (see Section 4.3.2), in order to provide a more robust likelihood model, the scores are treated as Dirichlet concentration parameters, and we sample likelihood values from the resulting Dirichlet distribution:

$$P(H_t \mid R_t) \sim \text{Dirichlet}(\alpha^R) \qquad P(H_t \mid A_t) \sim \text{Dirichlet}(\alpha^A)$$

4.4 Intent Inference

At each timestep, the DBN runs a prediction step followed by an update step.

Predict: The belief is propagated forward through the transition model:

$$\overline{\text{bel}}(R_t = r) = \sum_{r' \in \mathcal{R}} P(R_t = r \mid R_{t-1} = r') \cdot \text{bel}(R_{t-1} = r')$$
$$\overline{\text{bel}}(A_t = a) = \sum_{a' \in \mathcal{A}} \sum_{r \in \mathcal{R}} P(A_t = a \mid A_{t-1} = a',\, R_t = r) \cdot \text{bel}(A_{t-1} = a') \cdot \overline{\text{bel}}(R_t = r)$$

Update: The prediction is reweighted by observation likelihoods and renormalized:

$$\text{bel}(R_t = r) \propto \overline{\text{bel}}(R_t = r) \cdot P(G_t \mid R_t = r) \cdot P(V_t \mid R_t = r) \cdot P(H_t \mid R_t = r)$$
$$\text{bel}(A_t = a) \propto \overline{\text{bel}}(A_t = a) \cdot P(G_t \mid A_t = a) \cdot P(V_t \mid A_t = a) \cdot P(H_t \mid A_t = a)$$

Note that the action predict step sums over both a' and r to account for the coupling between referent and action beliefs introduced by the affordance mask. Additionally, gaze is continuous and contributes at every timestep, but voice and gesture contribute only during the single timestep in which they are inputted. $P(G_t \mid A_t)$ additionally

[7] For this task, we also use Gemini 2.5 Flash.

only applies during disambiguation, when actions have been assigned physical locations. When any modality is absent, we simply use a uniform vector for its observation node, which is mathematically equivalent to no input at all.

4.4.1 Intent Resolution

The system outputs its predicted target–action pair when both R_t and A_t have converged to a decision, and importantly, the user has made at least one voice or gesture input.[8] We define this as having one referent $r \in \mathcal{R}$ and one action $a \in \mathcal{A}_r$ each having crossed a minimum likelihood threshold of 0.85. We chose this value empirically, as it produced reliable decisions across our test scenarios without prematurely committing to an intent or requiring excessive additional input to resolve. Once this threshold is reached, the system immediately outputs (r, a) as its predicted intent.

4.4.2 Intent Disambiguation

The system often cannot resolve to a single intent following user input due to ambiguity. When this happens, the system communicates uncertainty to the user either visually or audibly. In most cases, this involves identifying which referents or actions are still *candidates* (defined as those meeting a minimum probability threshold) and allowing the user to clarify between them. Additionally, the system sometimes encodes the actual DBN belief as part of the user-in-the-loop strategy (e.g. scaling a candidate visually based on its likelihood).

Disambiguation mode is triggered when the user makes a voice or hand gesture input that is not immediately resolved into a single intent (i.e. at least one of bel(R_t) or bel(A_t) has not crossed the resolution threshold). The specific mode used is set as a

[8] This requirement of an additional input avoids the Midas Touch problem, where simply gazing at an object without any other input immediately selects it.

configurable system parameter. The system may have resolved only the referent, only the action, or neither. Because actions depend on referents in our framing, we allow the user to clarify in two sequential steps: referent disambiguation first, followed by action disambiguation if necessary. If the referent is already certain, the system skips directly to action disambiguation. This full user flow is illustrated in Figure 4.3.

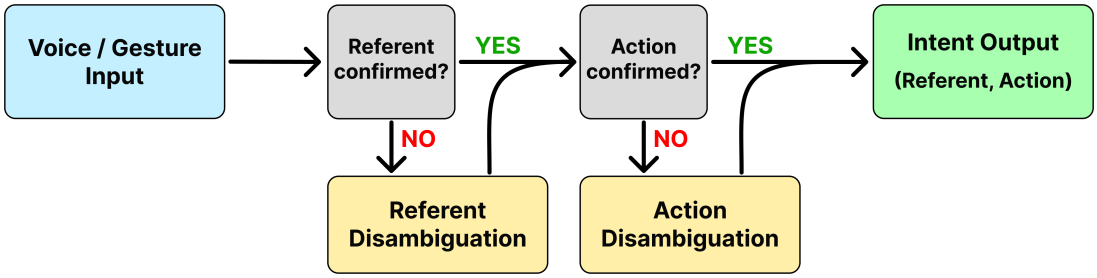


Figure 4.3: User flow of the disambiguation and resolution system.

Importantly, when the system enters disambiguation mode, it *freezes* the DBN's transition model, pausing temporal decay while still remaining responsive to new input observations. This, in effect, means that the network is no longer strictly *dynamic*. However, we retain the DBN framing since the full dynamic structure applies outside of disambiguation mode. This freezing is done to maintain the high probabilities of the top candidates, allowing the user to disambiguate quickly between a limited set of the most likely options while hiding any eliminated options.

While disambiguation mode is active, the user may continue to perform gaze, voice, and gesture inputs as before, and the DBN will continue to update likelihoods based on new observations without decaying. This continues until both $\text{bel}(R_t)$ and $\text{bel}(A_t)$ exceed the resolution threshold, at which point the system outputs its predicted target–action pair.

Chapter 5

Disambiguation Design

We design our system with the assumption that some user inputs are genuinely ambiguous, and even a perfect fusion system would struggle to resolve multimodal inputs that could reasonably map to several different interpretations. This ambiguity arises from two distinct sources: *sensor noise*, a technical limitation on input devices (rather than the fusion system itself) in which imprecise measurements obscure which object or action was intended, and *semantic underspecification*, an inherent property of language and gesture in which the input itself does not uniquely determine an interpretation regardless of measurement precision.

This introduces the problem of how best to communicate that uncertainty, and how best to allow the user to disambiguate between options. We design our disambiguation around two key principles: *one-bit disambiguation*, which asks the user to provide the minimum clarifying information needed to resolve uncertainty, and the complementary use of *identity* and *location*: surfacing identity to distinguish targets and assigning location to distinguish actions. To apply these principles, we explore several audio and visual modes of communicating uncertainty.

5.1 One-Bit Disambiguation

We design our system based on the concept of **one-bit disambiguation**, in which the user provides a single unit of information that clearly disambiguates their intent. This concept borrows loosely from information theory: the ideal disambiguation strategy maximizes the expected reduction in uncertainty with each clarifying input, thereby minimizing the required effort of the user (Reddy et al., 2022). The term "one-bit" describes a single clarifying input that is sufficient to identify the intended interpretation, regardless of the size of the candidate set.

This is especially relevant for ambient interaction contexts such as smart home environments, robotic control, and virtual assistants, where users are often multitasking and making imprecise or underspecified inputs. Interactions using imprecise pointing (Williamson and Murray-Smith, 2004; Guiard et al., 2004; Balakrishnan, 2004), gaze-based selection (Çığ Karaman and Sezgin, 2018; Xu et al., 2012; Mardanbegi et al., 2019; Lei et al., 2023), or underspecified speech (Chai et al., 2004; Lee et al., 2024; Hu et al., 2025) all produce ambiguity that an additional clarifying input can resolve.

5.2 Location and Identity

Disambiguation strategies must be carefully matched to the type of ambiguity. For this reason, we design target disambiguation and action disambiguation to be distinct, as each form is structurally different and calls for different display approaches (Lin et al., 2023). The key distinction lies in the roles of **location** and **identity**: targets have an intrinsic location but may lack a unique identity (e.g. multiple lights in the same vicinity are all simply "lights"), while actions have a unique identity but lack an intrinsic location (e.g. *brightness* and *color* are not spatial, but are easily identified by name). Our proposed uncertainty communication design in each case

compensates for what the input leaves unspecified: for targets, we hold location constant and assign identity; for actions, we hold identity constant and assign location. Through these additional audio and visual experiences, we aim to expand the set of possible disambiguation strategies by enabling new modalities or ways of referencing the intended referent or action.

5.2.1 Give Targets More Identity

For target disambiguation, we resolve ambiguity by giving each candidate target a distinguishable *identity*, so the user can identify the intended target when location is insufficient (e.g. when two candidates lie close in gaze). Target uncertainty generally occurs when candidates are densely packed, meaning location is no longer viable for low-effort disambiguation. Instead, we augment the targets with additional identifying information that allows users to disambiguate in other ways. For example, we display visual pointers over candidate targets, augment them with visual indicators of existing attributes of the object (e.g. color, relative size, relative position), or augment them with new identifiers like numerical labels. This newly surfaced information provides additional disambiguation tools across both voice and gesture modalities.

5.2.2 Give Actions Spatial Location

For action disambiguation, we resolve ambiguity by assigning each candidate action a spatial *location*, constructed by remapping actions as targets in the scene, so that the user can select the correct action when the input semantics alone are insufficient (e.g. when a gesture matches several candidate actions). When the input maps to multiple candidate actions, we enable additional disambiguation strategies by spatially embedding action targets within the scene. This can be done in several ways, such as plain text labels, an interactive UI, or targets embedded within the 3D scene. Users

can then select using spatial strategies like pointing, gazing, or verbally specifying a relative location, none of which would be possible without giving actions a location.

5.3 Target Disambiguation Design

In target disambiguation, uncertainty is communicated to the user in one of four modes, each differing in what identifying information is presented. The first mode is purely auditory, while the remaining three modes are visual. The audio mode cannot communicate candidates in any way, since their identities may be ambiguous and unable to be distinguished by name alone.[1] It merely informs the user that the system is uncertain. The visual modes, on the other hand, are able to communicate the candidate targets by displaying a small overlay anchored directly above each candidate object in the scene. Each mode imposes a different type of identifying information on the targets, ranging from simply marking their presence, to surfacing existing attributes, to introducing new assigned identifiers, so the user can see which candidates have been selected and how to refer to them. The four target disambiguation modes are shown in Figure 5.1 and described as follows:

(1) Audio. The system outputs audio that conveys uncertainty in the target. Specifically, it says, *"Which one?"* It does not provide any additional information.

(2) Visual – Pointers. The system displays visual pointers above the candidate objects, marking their presence without providing any further identifying information.

(3) Visual – Attribute Labels. The system displays visual markers above the candidate objects, which (if applicable) convey some intrinsic attribute of the object, such as its color, relative size, or relative position. We use *color* as the intrinsic

[1] We design for the worst case in which candidate targets share the same semantic label (e.g. multiple "lights"). In practice, targets may sometimes be distinguishable by name.

1. Audio

2. Visual – Pointers

3. Visual – Attribute Labels

4. Visual – Index Tags

Figure 5.1: The four target disambiguation modes, each displayed after the system was unsure which of five books the user meant to select.

attribute here. These markers are meant to indicate to the user that they may use this attribute to disambiguate via voice input.

(4) Visual – Index Tags. The system displays visual numbered index tags above the candidate objects. This is distinct from attribute labels in that these numbers are not *intrinsic* attributes, and the system assigns them *arbitrarily* for the purposes of disambiguation. The tags are meant to indicate to the user that they may use the corresponding number to identify the object via voice or through an iconic gesture.[2]

[2]An iconic number gesture uses extended fingers to represent a number.

5.4 Action Disambiguation Design

In action disambiguation, uncertainty is communicated to the user in its own set of four modes, distinct from those used for target disambiguation, each differing in layout representation and afforded interaction modalities. As in target selection, the first mode is purely auditory, but the action audio mode *is* able to communicate the candidate actions, as it can identify them directly by name. The three visual modes, in accordance with our design philosophy of making actions into *spatial targets*, each embed the actions within the scene differently: displaying text of the candidates, presenting a menu of direct selection buttons, and placing action targets within the scene itself to enable gaze and pointing. This progression reflects an increasing degree of spatial embedding, trading simplicity for richer interaction affordances. The four action disambiguation modes are shown in Figure 5.2 and described as follows:

(1) Audio. The system outputs audio that conveys uncertainty in the action, and lists the options for the user to choose from. Specifically, it says, *"Did you mean X, Y, or Z?"*, where X, Y, and Z represent candidate actions. The user is meant to respond via voice.

(2) Visual – Text. The system displays text in a small window that lists the candidate actions. Specifically, it says, *"X, Y, or Z"*, where X, Y, and Z represent candidate actions. The user is meant to respond via voice.

(3) Visual – Button Menu. The system displays a vertical list of rectangular buttons, each labeled with one of the candidate actions, appearing in front of the selected referent object. Buttons are selectable by casting a ray with the hand or controller and performing a pinch or poke gesture, immediately selecting the action.[3]

[3]A pinch brings the index finger and thumb together (like a click); a poke extends the index finger to touch a surface. Both are standard Meta Quest interaction primitives.

1. Audio

2. Visual – Text

3. Visual – Button Menu

4. Visual – Spatial Targets

Figure 5.2: The four action disambiguation modes, each displayed after the system was unsure whether a *swipe up* gesture at a lamp was indicative of *brightness up* or *hue up*.

(4) Visual – Spatial Targets. The system displays circular targets labeled with the candidate action names, arranged radially around the referent object in the scene. This mode introduces the gaze modality for action selection. Gaze functions here as a direct selection mechanism where the user dwells on an action target until it is selected, with the target providing continuous visual feedback on progress by physically growing in size and shifting from white to green as its likelihood increases. The targets also allow for pointing gestures to identify actions in space, which work the same as pointing to referent targets.

Chapter 6

User Study

We conducted a user study to evaluate our multimodal fusion system and our disambiguation designs. We separated these evaluations into two different phases that isolated each part: an elicitation study where we collected users' natural multimodal inputs as validation of our system design, and a comparative study where we analyzed user feedback on each of our disambiguation designs. We recruited 8 undergraduate students at Princeton University (6M, 2F; ages 18–22) via convenience sampling, compensating each with $16 for their time. This study was approved by Princeton University's Institutional Review Board (IRB #16692) and all participants provided informed consent before participation. For each participant, we ran Phase 1 and Phase 2 in sequence.

6.1 Phase 1: Multimodal Elicitation

The first phase of our study was an elicitation study in which participants interacted with a speaker and a lamp, positioned on a table in front of them, as shown in Figure 6.1. Participants were told that they could interact with the objects via voice

or gesture, and that their head orientation would also be taken into account. The goal of this part of the study was to collect a dataset of *which* specific inputs participants performed, in order to evaluate the system on what they naturally did. For this reason, we disconnected the main fusion system during the study session and used a Wizard-of-Oz system so that participants would always be "successful" in their input attempts, meaning that after any input, we manually invoked the intended action behind the scenes to simulate the system working (e.g. the light turning on), regardless of whether it would have succeeded in the real system. We did this because the main goal was to collect data about which input methods and strategies they used, without being influenced by any feedback from the system.

Figure 6.1: The study scene that participants interacted with across Phase 1 and Phase 2, containing a lamp, a speaker, and several books.

6.1.1 Elicitation Tasks

We split the elicitation into twelve tasks: six involving the lamp and six involving the speaker. Each participant completed all twelve tasks in each of two distinct study conditions, for 24 total task trials per participant. In each task, we instructed users

to trigger a single distinct action on the corresponding interactive object. Before beginning, participants completed two practice trials (one voice and one gesture) on neutral tasks to familiarize themselves with the modalities without priming any of the actual study gestures or commands. Practice trial data was not logged.

Rather than describing the goal in written or spoken words (e.g. *"turn on the lamp"*), which may have inadvertently influenced the user's chosen input method, we communicated the intended goal through a visual "flashcard" depicting only the start state and end state (e.g. a lamp in its *off* state and *on* state). For the speaker cards, we included universally recognizable symbols that clearly described the state without priming the participant toward a specific input method. After receiving feedback from colleagues on the lamp cards, we also decided to add symbols to depict the relative brightness and color (since it was sometimes unclear whether the goal was to make it dimmer/brighter or make it warmer/cooler). We wanted to make each goal as clear as possible and keep that understanding constant across participants to allow us to observe the differences in how participants chose to achieve that goal. All twelve flashcards are displayed in Figure 6.2.

6.1.2 Modality Constraints

Phase 1 was divided into two study conditions, and participants ran through all tasks in each condition sequentially. During the first condition (**free-form**), participants were allowed to perform whatever input they wanted, and were specifically instructed to perform the input that would accomplish the task in the simplest way. In the second condition (**voice-constrained**), participants were not allowed to use voice input, and had to use only gestures. We did this because we believed users may otherwise default to using voice only, as it can generally provide a clear and unambiguous input for these tasks. We motivated this decision by noting that real-life scenarios often

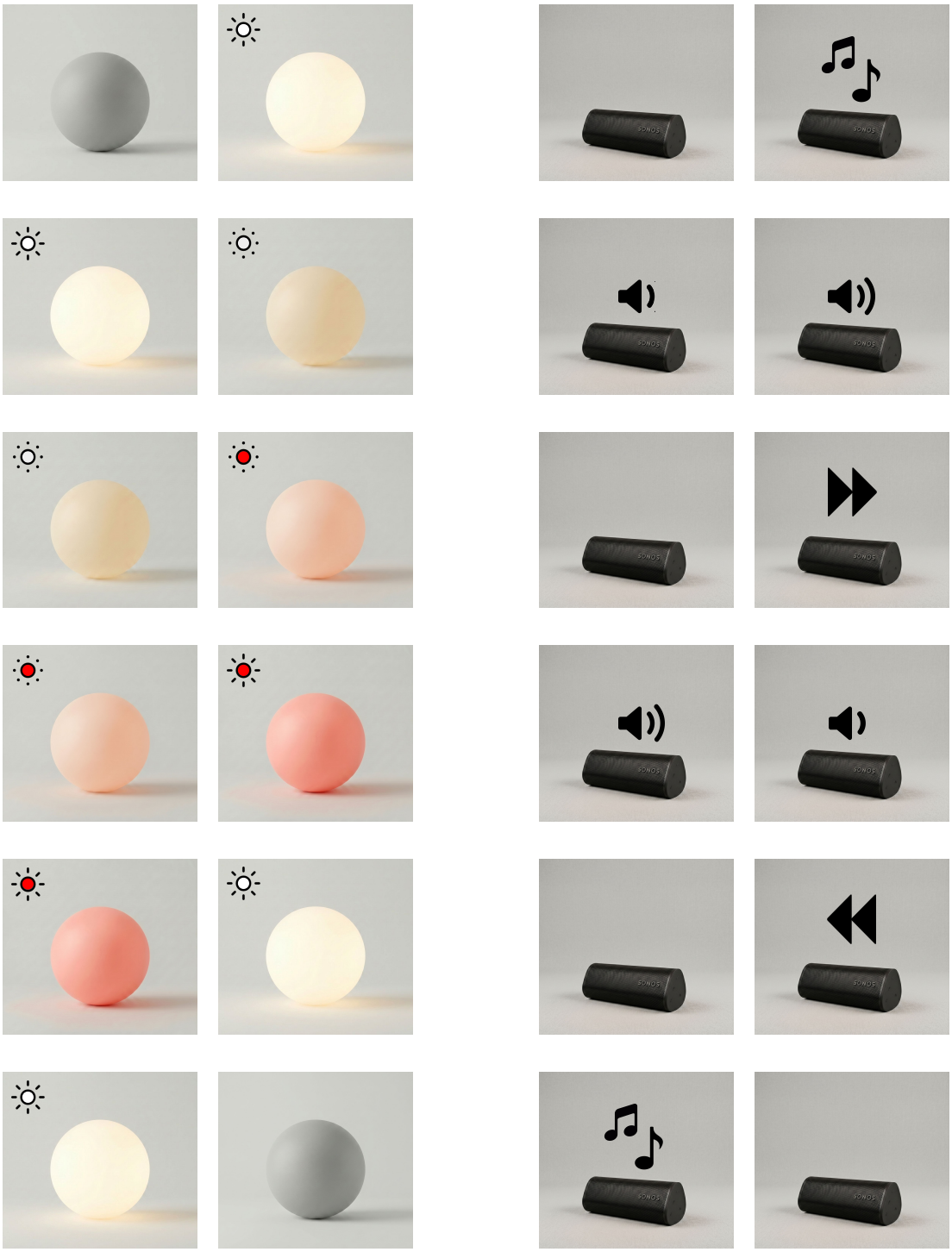


Figure 6.2: The flashcards used in the elicitation study. Each pair of images is one flashcard, where the left image describes the start state and the right image describes the end state. The goal of each task is to get the object into its end state.

restrict available modalities, such as in a crowded train, a loud concert, or a quiet library, where voice may be unavailable or inappropriate. To counterbalance ordering effects, we randomized the condition order across participants, where half completed the free-form condition first and half completed the voice-constrained condition first. Within each condition, we also randomized whether participants completed the lamp block or the speaker block first. The order of tasks within each object block was fixed, since the tasks are state-dependent and must be performed in sequence.

6.1.3 System Validation

During both conditions of Phase 1, rather than providing direct feedback through the AR interface, we simply *logged* the user's state and current inputs at each frame and stored them on our server. After the user performed any input for a task, we fed the raw input data into an offline version of our system and later measured whether it correctly executed the intended action. We always invoked the intended action during the live session to maintain the Wizard-of-Oz simulation.

6.2 Phase 2: Disambiguation Comparison

In the second part of the study, we compared user interactions within the disambiguation design space proposed in Chapter 5. We separated the phase into target selection and action selection to study the disambiguation strategies separately. As established before, each case contains four distinct modes of communicating uncertainty: one audio and three visual, presented in counterbalanced order across participants. In each case, we isolated the relevant disambiguation task (target vs. action) by using different scene setups. Namely, we used a row of books each with a single action for target selection, and we used a single lamp with a number of actions for action selection. Note also that Phase 2 used our full fusion system described in Chapter 4 and was *not*

Wizard-of-Oz. We intentionally did not provide practice trials prior to Phase 2, as we sought to capture participants' initial, unprimed reactions to each disambiguation mode. Unlike in Phase 1 where we communicated the goal but allowed the user to perform an input using any method they wanted, here we specifically instructed the user to perform an exact input, constant across all tasks, that would always cause the system to enter disambiguation mode. This controlled for the initial input and allowed us to focus our attention on how the user interacted with the uncertainty visualization, determining whether they could successfully disambiguate (i.e. the correct action being executed).

6.2.1 Target Selection Comparison

The target selection tasks used a row of five books positioned side-by-side on a table in front of participants so that only their spines were visible. Each book afforded a single action, *select*, which served only to identify the book rather than trigger any visible effect.[1] For each of the four tasks, we instructed participants to select a specific book called *Human-Centered AI* by looking toward the book and speaking the word *"select."* This caused the system to immediately enter disambiguation mode as it was uncertain which book the user was referring to, since all five likely maintained nonzero probabilities given the imprecise gaze input toward the clustered group of targets. Each of the four tasks used a different target disambiguation mode, presented in counterbalanced order across participants, as described in Section 5.3. We observed how users responded to each mode.

[1]Here we relax the assumption that the interactive objects need to be "smart" in the traditional sense. You may think of the *select* task as being equivalent to an AI assistant identifying the book's name or how much it costs. For more discussion on this, see Chapter 8.

6.2.2 Action Selection Comparison

The action selection tasks used a single lamp positioned on the left side of a table in front of participants. The lamp afforded five actions: *toggle*, *brightness up*, *brightness down*, *hue up*, and *hue down*. For each of the four tasks, we instructed participants to increase the brightness of the lamp by looking at it and performing a *swipe up* gesture. This caused the system to immediately enter disambiguation mode as it was uncertain whether the user was trying to invoke the *brightness up* or *hue up* action, since both actions were associated with a continuous upward motion. Each of the four tasks used a different action disambiguation mode, presented in counterbalanced order across participants, as described in Section 5.4. We observed how users responded to each mode.

6.2.3 Feedback Collection

After each disambiguation task, we asked participants to answer two Likert-scale questions about their experience, namely how easy or difficult it was to resolve the system's uncertainty, and to what degree they felt in control of the interaction. We also asked them if any part of the interaction was unclear or frustrating. At the end of all target selection tasks and at the end of all action selection tasks, we asked participants which of the four modes they preferred for disambiguating. This feedback aimed to assess the feasibility of the design space and inform future design work.

Chapter 7

Evaluation

We evaluate our study in several parts. First, we analyze the voice and gesture inputs produced by the elicitation study in Phase 1 and examine the associated user interaction patterns. We then evaluate the accuracy of our multimodal fusion system using the collected inputs. Next, we present participant feedback on each disambiguation design from Phase 2 and compare their tradeoffs. Finally, we identify several takeaways from the study to inform future disambiguation designs.

7.1 Elicitation Results

We were first interested in which modalities participants naturally used in the free-form mode, when both voice and gesture were allowed. 4 of 8 participants always used both modalities simultaneously, while 3 used voice alone, and only one (P3) used different modalities for different trials. In the voice-constrained mode, all participants were forced to use gestures. For most of the actions in the scene, including volume controls, brightness controls, and discrete on/off actions, users generally were able to intuit a gesture that mapped to the action in a semantically meaningful way. Users

naturally gravitated toward dynamic, directional gestures for continuous actions (e.g. swiping motions, oscillations) and snaps, pinches, or fists for discrete actions.

Participants consistently noted that the speaker was much simpler to control via gesture than the lamp was, since its afforded actions map naturally to gesture. Most lamp interactions relied on voice in the free-form condition (with users speaking commands such as *"turn on," "dim," "red light," "darker red," "go back to white,"* etc.), and users often got stuck in the voice-constrained condition using gestures alone. P1 reported that they tried to isolate the lamp actions into distinct "dimensions" (i.e. brightness and hue) using different directional gestures, namely vertical motion and angular motion (i.e. rotating their wrist as if turning a knob). P8 similarly decided to map hue onto a rotational axis, tracing a circular path with their whole hand. Almost all participants expressed frustration with changing the lamp's hue via gesture, finding no natural motion or symbol that mapped onto the concept of color. The range of resulting input strategies was vast: P2 wrote letters mid-air with the index finger ("R" and "W" for *red* and *white*); P6 signed the words *red* and *white* in ASL; and most strikingly, P3, P4, and P5 each independently pointed to a red book that happened to be in the scene, using it as an external referent to communicate the color *red*. These varying responses validate the need for free-form gesture recognition that we describe in Chapter 3.

Most participants said that they preferred voice as an input method. In general, they felt that the voice-constrained condition made it considerably more difficult to express intent. Still, many noted that hand gestures might be better for certain tasks or in specific contextual situations. For instance, P6 said that gestures were more convenient for simple tasks than having to speak a voice command, but for more complex tasks, voice better captured intent. P8 noted that directional gestures felt very natural, as if one were implicitly manipulating user interface elements like sliders

that have conventional directions (e.g. volume implies vertical motion, skip/previous implies horizontal motion). P8 also noted that gestures would be better in certain contexts, such as when alone at home and feeling awkward speaking, or in public where speaking aloud is undesirable. P1 mentioned that gestures could be done more discreetly without making noise, and existing tactile interactions (they specifically noted swiping on AirPods) translated seamlessly to some of these mid-air interactions, making gesture feel more natural.

7.2 Fusion Validation

We recorded multimodal inputs during Phase 1 and replayed them through our fusion system in offline mode to evaluate whether the system correctly resolved the intended action. For the free-form condition (i.e. voice and gesture both allowed), the system correctly resolved to a single intent in 53.1% of trials, and the correct action was present in its resulting *candidate set* in 83.3% of trials. However, for the voice-constrained condition (i.e. gesture only), the system correctly resolved to a single intent in just 7.3% of trials, and the correct action was present in the candidate set in 80.2% of trials. In cases where the correct action was in the candidate set, the system was not confident enough to output an intent with certainty, including trials where nothing was pruned at all. Note that since the validation was run offline without live feedback, disambiguation mode could not actually be triggered. As a result, these results only reflect users' initial multimodal inputs, not the full interactive loop.

Our results support and validate our multimodal fusion system and the use of the Dynamic Bayesian Network for inference. It should be noted that our 53.1% of multimodal trials correctly resolving means that most interactions would have succeeded in one step. In a live session, the remaining trials that did not resolve would have triggered disambiguation mode. As evidenced by the 83.3% of trials with the correct

intent in the candidate set, the system is often uncertain, but rarely confidently incorrect. This was an intentional design choice for our system: to be slightly conservative in prediction and allow the user to take that extra step of clarification.

Additionally, the gap in accuracy between the free-form condition and the voice-constrained condition is substantial, showing that multimodal input vastly outperforms gesture alone. Voice input was consistent and reliably steered the system toward the correct action. Conversely, the low accuracy for the gesture-only condition reveals that gesture heuristics alone are insufficient to capture the wide variety of gestures users produce without specific instructions, even when the gesture types were designed around the objects and actions in the study. This is not strictly a limitation of the fusion system, but a limitation of the gesture recognizer. As discussed in Chapter 3, our heuristics model is a substitute for what should be a much more capable free-form gesture processor, and users often produced gestures that our system could not feasibly recognize, such as directional oscillatory motions.

Some errors arose from task misinterpretation or ambiguous flashcard descriptions (e.g. users often interpreted a card intended to change the lamp to a "brighter red," mapping to *brightness up*, as "more red," mapping to *hue up*). In these cases, the system acted correctly according to the user's input, but incorrectly according to the ground truth for the task. Other errors arose due to problems with the input trigger, i.e. starting and stopping the trigger too early or too late. This led to some voice inputs being accidentally cut off or omitted, or gesture recordings containing additional motions that were not strictly part of the meaningful gesture. All of these are limitations of our system's interaction paradigm and our study design.

Despite promising results for our fusion system itself, offline validation captures only the first pass of a system designed around iterative clarification, and the disambiguation study that follows evaluates the full interactive experience of the system.

7.3 Disambiguation Comparison

For each of the four target and four action disambiguation modes, participants rated their experience on two five-point scales: **ease of disambiguation** (1 = difficult, 5 = easy) and **level of control** (1 = low, 5 = high). We also collected open-ended feedback on which modes felt frustrating or unclear, and asked participants to identify their preferred mode overall. Across all modes, participants succeeded in their disambiguation interaction on the first try (i.e. the system resolved to an intent after one input) in 53 of 64 trials (82.8%), and all participants eventually got the system to resolve on all trials following subsequent attempts.

7.3.1 Target Disambiguation

For target disambiguation, 5 of 8 participants preferred **audio only**, with a mean ease of disambiguation score of 4.9 (see Figure 7.1). In this mode, all participants successfully disambiguated after the first attempt. Despite the mode surfacing no additional information to facilitate disambiguation, users appreciated not having to interact with additional UI elements. Most participants clarified the specific referent by identifying its color via voice, saying it was the most immediately obvious attribute of the book to distinguish it. Others used its position in the set (e.g. "the fourth one"), with P1 even performing a four-finger gesture simultaneously. However, participants also noted that this was only easy because the books were already visually distinguishable (e.g. via color), and they were skeptical that an approach like this would work at scale.

Results were almost identical between the **attribute labels** (i.e. color labels) and **index tags** modes. For both of these modes, all participants successfully disambiguated after the first attempt. We separated these conditions because we wanted

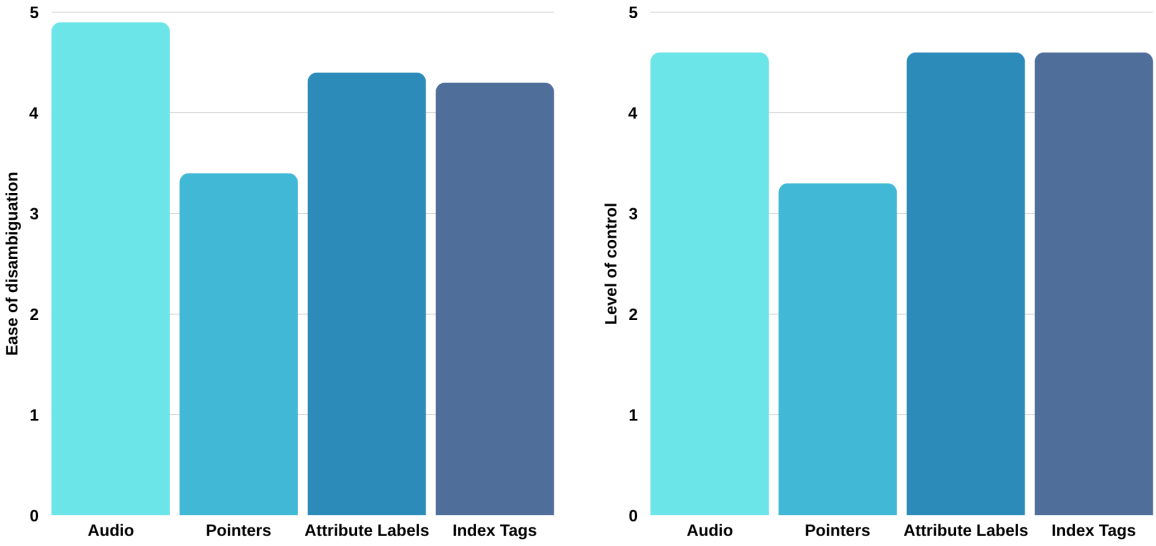


Figure 7.1: Ease of disambiguation (left) and level of control (right) ratings across target disambiguation modes.

to determine whether surfacing an *existing* attribute (color) would have a different effect than surfacing a *new* identifier (an index number). The two conditions had roughly the same effect. Because each of these essentially served the same function in uniquely identifying each of the books, it followed that users perceived the interactions in the same way, just with a different identifier. It is worth noting, however, that surfacing a new identifier would likely outperform surfacing an existing attribute in a case where the objects have no distinguishable attributes intrinsic to them. Every one of our participants spoke the name of the presented identifier in order to disambiguate. Some users, such as P5, did not like how these modes essentially forced a specific disambiguation strategy on them, wishing they could decide for themselves how to identify the books rather than conforming to the presented visual identifier.

The most controversial mode was visual **pointer** mode, which received a mean ease of disambiguation score of 3.4 (SD = 1.5), as users felt it introduced unnecessary UI complexity without providing any additional useful information. Only 5 of 8 participants successfully disambiguated after the first attempt. P3 said it was frustrating

to interpret a foreign UI system instead of expressing the book's identity directly via voice. Interestingly, the disambiguation strategy elicited by this mode was different among participants: some believed the system was encouraging them to *point* at the intended referent (which did not resolve since the books were too close together spatially and pointing is imprecise), while others verbally identified the order of the pointer in the set (e.g. "the fourth one" or "the second from the right"), and still others simply identified the book's color, as before.

7.3.2 Action Disambiguation

For action disambiguation, 5 of 8 participants preferred the visual **button menu**, reaching a mean ease of disambiguation score of 4.8 (see Figure 7.2). All participants immediately understood the raycast interaction and successfully disambiguated after the first attempt. Participants liked the button menu because it reminded them of interfaces that were familiar to them, as the Meta Quest's raycast selection was intuitive and natural to some, but also simply because it resembled a clear menu from which you could select a single button (some even compared the visual display to a Wii interface). The only downside to the button menu was its large size, as several participants mentioned it obscured the referent object (the lamp).

As with target disambiguation, users also rated the **audio** mode highly, since it felt most natural to respond to a spoken prompt with speech, and all participants successfully disambiguated after the first attempt. The **text** mode, which communicated the exact same information but visually rather than audibly, was slightly lower in score than audio, with 6 of 8 users successfully disambiguating after the first attempt. Users responded to the audio mode with voice every time, saying that it was more intuitive to respond to speech with more speech. For the text mode, most users responded with voice, but some tried other things first (P6 tried ASL again), and generally felt

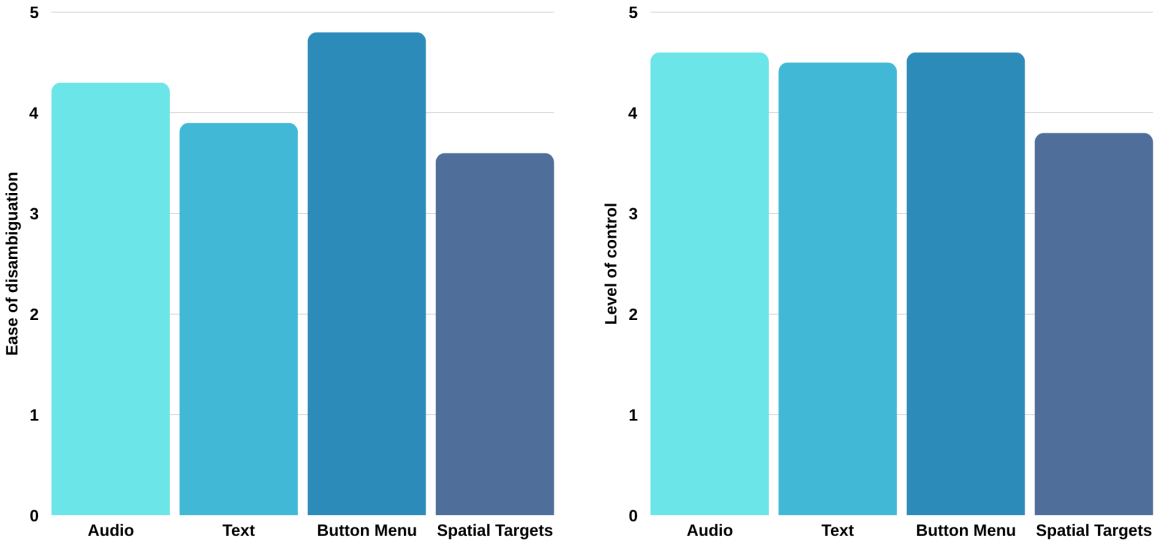


Figure 7.2: Ease of disambiguation (left) and level of control (right) ratings across action disambiguation modes.

the text display did not suggest a clear response modality. P5 felt the text prompt resembled an error message, and P1, P3, P4, and P8 explicitly noted they would prefer listening to audio over reading text. P7, on the other hand, said they preferred text because the audio dictation took too long and felt robotic.

Finally, most participants found it difficult to interact with the **spatial targets** mode, with a mean ease of disambiguation score of 3.6. Only 2 of 8 participants successfully disambiguated after the first input attempt. This was the most novel mode, using gaze dwell as the primary interaction mechanism, which proved to be unintuitive to users upon first interaction. Some understandably thought the targets functioned as physical buttons they were meant to press or touch (similar to the button menu mode) and expected the raycast visual to appear. Most participants noticed the targets increase in size and become greener, but only 2 participants understood that gaze was the mechanism that caused the effect. However, all participants eventually did select the correct action, albeit with some delay, mainly because their attempts at interacting with the targets via other means caused them to inadvertently gaze

toward them, registering an input signal despite most participants not understanding the mechanism. This suggests that gaze dwell is not an intuitive interaction method without explicit instruction, but still raises the question of whether gaze-based spatial targets could be viable disambiguation methods if prior instructions were given.

7.4 Design Takeaways

Based on the results of our comparative study, we identify several patterns that can inform disambiguation designs going forward.

First, the uncertainty modality influences the response modality. Participants naturally responded to audio messages with voice inputs, but were more hesitant to speak when confronted with visual information. The visual modes tended to facilitate gestures, with users naturally reaching to point to visual elements (especially in the pointer target disambiguation mode), despite the imprecision of pointing making the referent unresolvable. This was also seen in action disambiguation, as all users immediately understood to press the buttons in the menu via raycast, but attempted to point or gesture toward the spatial targets, which failed to resolve due to the imprecision of pointing and the close spatial proximity of the actions. The gap between what a mode *suggests* and what it actually *supports* directly affected user satisfaction. Broadly, any disambiguation mode that implies a strategy that it cannot actually deliver will cause frustration.

Another major takeaway is that memory and familiarity play a role in what users like best. Multiple participants expressed that the button menu felt intuitive and obvious because it resembled traditional interfaces that they were already used to, and that the audio modes were unambiguous and direct. Conversely, the gaze-based spatial targets performed much more poorly due to their unfamiliar interaction paradigm.

New interaction types like this require onboarding before being useful. Given explicit instructions or even customization, it may be possible to design even better disambiguation methods that specifically make use of muscle memory rather than assuming no prior use.

Finally, participants were largely split on one major design choice: how much the audio or visual mode *restricted* the available disambiguation strategies. Some users preferred the system specifying exactly one identifier or explicitly asking for clarification rather than being more flexible and open to interpretation. P2, P4, and P8 noted that modes like index tags, which restricted response options by directly labeling targets, made the expected action clear and unambiguous. P5 liked that the button menu especially felt like it "stopped" them before proceeding to ensure it understood their intent. However, P5 also noted that the color labels put pressure on them to figure out the system's logic rather than expressing intent naturally. P3 and P7 disliked having to interpret arbitrary UI choices and map them mentally to their intended function, rather than expressing their intent in their own words. In summary, forced disambiguation reduces cognitive load for users who find open-ended response ambiguous, but reduces perceived agency for users who prefer expressive freedom. The optimal system would be able to do both: offer structured, unambiguous options while also accepting free-form responses.

Chapter 8

Applications

At a high level, our system allows a user to provide imprecise multimodal inputs to trigger a specific action associated with an object in their environment, and disambiguate between objects and actions to remedy that imprecision. This problem framing has a range of applications, from smart home interaction to robotics and more. In this chapter, we explore the design space of agents acting in the physical world by identifying three main categories of application that would benefit from a multimodal fusion system with user-in-the-loop disambiguation techniques: embedded agents, embodied agents, and disembodied agents.

Crucially, the disambiguation interaction design is invariant across all three categories, as candidates can be surfaced spatially in AR and resolved through the same multimodal inputs regardless of whether the executing agent is embedded in an object, embodied in a robot, or disembodied in a virtual assistant. In each case, we define the agent category, describe how our system maps onto it, and show examples of how this might work in the real world. This paradigm for user-in-the-loop disambiguation of agent actions stands as a great addition to existing HRI and HAI frameworks.

8.1 Embedded Agents

The primary use case for the system that we have explored so far in this thesis is smart home control, which we will broadly classify as *embedded agents*: computation built into physical objects. This is the natural application for the problem because our model of objects and afforded actions maps naturally onto the specific intents afforded by smart home objects. The target–action pair is simply an IoT device and a specific affordance for the device to execute. In other words, the computation lives in the object and the manipulation is done by the object.

(a) (b)

Figure 8.1: (a) Target disambiguation between lamps using numerical tags. (b) Action disambiguation for a robot vacuum.

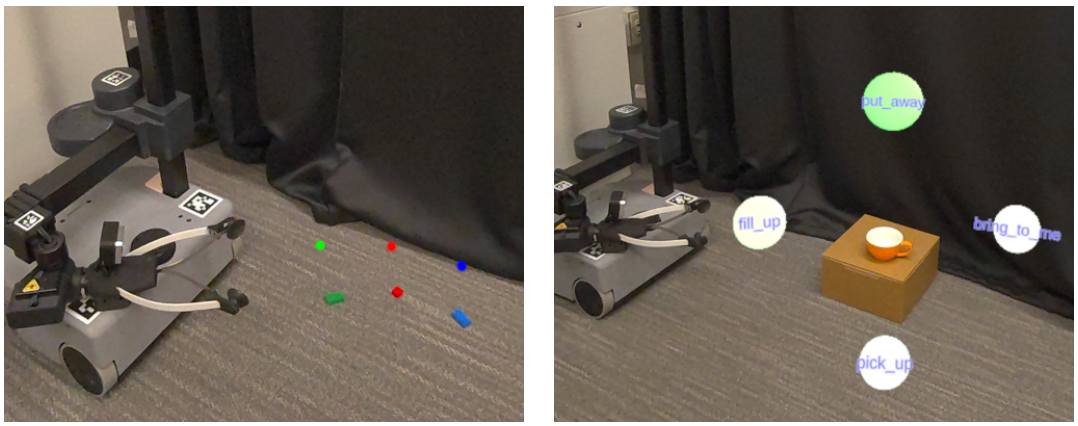
We explore embedded agent examples in depth in our action selection scenario in Section 5.4 and our elicitation study in Section 6.1, which make use of a smart lamp and a smart speaker, each with a variety of actions. Shown in Figure 8.1 are two more examples of embedded agents using our system, including a target disambiguation between two identical lamps, and an action disambiguation for a robot vacuum.[1] On

[1] A robot vacuum is technically an embodied agent because it moves around and cleans the floor, but by our definition it is also embedded. This is because the user commands the device itself rather than directing it to act on a separate passive object, as is the case for the robot in Section 8.2.

the left (a), the user spoke, *"turn on the lamp,"* leading to an ambiguous referent, activating target disambiguation. The user could disambiguate by speaking the number or performing an iconic number gesture. On the right (b), the user waved at the vacuum, but the action was ambiguous between *dock* and *clean*. Here, the user could disambiguate by gazing or pointing at the intended action.

8.2 Embodied Agents

Our system naturally extends from embedded agents into the realm of *embodied agents* (i.e. robots): agents with a physical presence in the world that can manipulate it. Instead of specific actions being afforded by objects, a robot performs specific actions *on* or *to* the objects. In this case, the target–action pair is an object to interact with and a task for the robot to accomplish using it. This is distinct from the *embedded* case because here, an *external* agent (i.e. the robot) is the one actually executing the task, rather than the object itself. In other words, the computation is separated from the object of interest and so the manipulation is done by an external agent.



(a) (b)

Figure 8.2: (a) Target disambiguation between LEGO bricks using colored markers. (b) Action disambiguation between robot tasks involving a coffee cup.

As household robots become more capable, users will need to direct them through natural and imprecise multimodal cues like pointing, gesturing, and speaking, rather

than through predefined commands or controllers. Prior work has established frameworks for detecting when robot commands are ambiguous (Ren et al., 2023; Park et al., 2024) and generating clarifying questions via language. Our system complements this by providing the interaction design for *how* that clarification is sought, surfacing candidate interpretations visually in AR and enabling intent resolution. This treats disambiguation not as a failure, but as a natural artifact of imprecision.

Figure 8.2 shows how robot task disambiguation might look in practice[2], including tasks involving LEGO bricks and a coffee cup. On the left (a), the user spoke, *"pick up the brick,"* leading to an ambiguous referent. In this case, the system encouraged the user to disambiguate by speaking the color of the brick. On the right (b), the user pointed at the cup, prompting action disambiguation among the robot's four possible actions for the cup: *pick up*, *fill up*, *put away*, and *bring to me*. The user could gaze or point to disambiguate. Notice again that these are actions the robot performs in relation to the cup, not actions done by the cup itself.

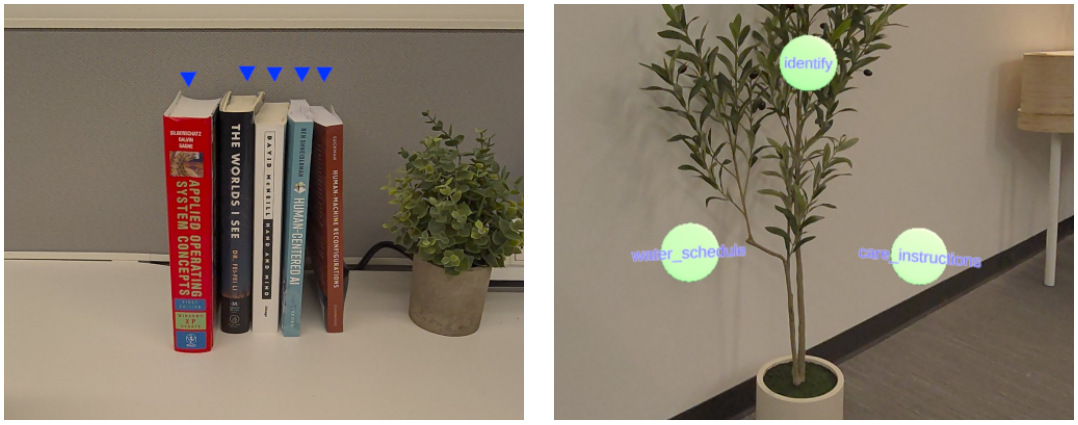
8.3 Disembodied Agents

Finally, our system extends to *disembodied agents*: virtual agents that can perceive the physical world and respond informationally, but cannot physically manipulate it. Unlike smart home devices or robots, a disembodied agent leaves the world unchanged, as it only reasons and responds. In this case, the target–action pair is a referent object and some information from the virtual agent about that object.

A system like ours might be useful in this category for smart glasses, for instance, so that users could ask a built-in AI assistant questions about specific things they

[2]These figures were created by running the system in a simulated environment with a robot present in the scene; the system was not actually connected to or controlling the robot.

see in the world around them. The agent would disambiguate targets in AR to determine what object the user is referring to, and disambiguate actions to determine what the user would like to know or accomplish with that object. Systems like GazePointAR (Lee et al., 2024) already demonstrate this pattern, using gaze and pointing to disambiguate pronoun references so that an AI assistant can answer questions.



(a) (b)

Figure 8.3: (a) Target disambiguation between books using basic pointers. (b) Action disambiguation between AI assistant responses involving a plant.

The book disambiguation task from Section 5.3 is an example of a disembodied agent. Figure 8.3 shows the book setup again for target disambiguation, along with a plant for action disambiguation. Notice that neither of these scenes contains "smart" or interactive objects, nor is there a robot in the scene to manipulate them. Both cases involve the user referencing a physical object while invoking a virtual agent. On the left (a), the user spoke, *"what was the title of that book?"* once again leading to an ambiguous referent and activating target disambiguation. The user is meant to disambiguate using an intrinsic attribute of the book, such as color, size, or relative position. On the right (b), the user pointed at the plant, but gave no indication of the intended action, leading to action disambiguation between the virtual agent's three plant-related actions: *identify*, *water schedule*, and *care instructions*. Here, once again, the user can disambiguate by gazing or pointing at the intended action.

Chapter 9

Conclusion

We built a multimodal fusion system that accepts gaze, voice, and gesture input in AR, uses a Dynamic Bayesian Network to predict the most likely target–action pair, and triggers user-in-the-loop disambiguation in cases of uncertainty. Our results indicate that multimodal inputs are intuitive and preferred, and that users favor simple, familiar disambiguation interactions such as audio for targets and a visual button menu for actions, while novel paradigms like gaze-based mechanisms are harder to understand. We observed that the modality used to surface uncertainty influenced how users responded: audio prompts naturally elicited voice responses, while visual displays tended to encourage gesture or pointing attempts. Finally, we identified a fundamental tension between how much control users want over their disambiguation strategy and how much the system should handle on their behalf.

The most significant limitation was the absence of free-form gesture recognition, which limited the system to a predefined vocabulary and left open gesture inputs unrecognized. Gaze inputs were also limited by the capabilities of the Meta Quest, which does not support eye tracking, so we approximated gaze using head position. Our

study was limited to 8 participants, and a larger sample size might have uncovered clearer trends or additional nuances. The setup itself was also constrained to a lamp, a speaker, and some books, and the specific tasks presented limit generalizability to more complex or varied environments. Lastly, the fusion validation was conducted offline, without the interactive disambiguation loop that characterizes real use.

Future work should address each of these limitations. Fully integrating an open gesture recognition pipeline would replace the heuristic vocabulary and enable free-form input. Integrating true eye gaze tracking would enable more precise selection. A larger user study across more varied environments would surface clearer trends and improve generalizability. Further evaluation across the application scenarios outlined in Chapter 8, particularly robotic control, would also be beneficial. In addition, future work may take into account the role of memory and learning effects in disambiguation modes, to determine if other modes besides the ones users preferred in our study could actually be better given a baseline muscle memory. Beyond addressing these limitations, several extensions would further enrich the system. For instance, a *reset* capability would allow users to restart disambiguation if the candidate set is wrong, and an *undo* capability would allow them to reverse a completed action. Finally, systems should model *risk* across actions, weighting predictions by consequence and requiring explicit confirmation for high-stakes operations.

Overall, we find that disambiguation is a natural and intuitive part of multimodal AR interaction, and well-designed audio and visual feedback mechanisms can make clarification feel effortless. We hope this work provides a foundation for that design space, where users can resolve their intent with ***a bit of clarification***.

Bibliography

Ravin Balakrishnan. "beating" fitts' law: virtual enhancements for pointing facilitation. *Int. J. Hum.-Comput. Stud.*, 61(6):857–874, December 2004. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2004.09.002. URL https://doi.org/10.1016/j.ijhcs.2004.09.002.

Richard A. Bolt. "put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, page 262–270, New York, NY, USA, 1980. Association for Computing Machinery. ISBN 0897910214. doi: 10.1145/800250.807503. URL https://doi.org/10.1145/800250.807503.

Çağla Çığ Karaman and Tevfik Metin Sezgin. Gaze-based predictive user interfaces: Visualizing user intentions in the presence of uncertainty. *Int. J. Hum.-Comput. Stud.*, 111(C):78–91, March 2018. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2017.11.005. URL https://doi.org/10.1016/j.ijhcs.2017.11.005.

Joyce Y. Chai, Pengyu Hong, and Michelle X. Zhou. A probabilistic approach to reference resolution in multimodal user interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, IUI '04, page 70–77, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138156. doi: 10.1145/964442.964457. URL https://doi.org/10.1145/964442.964457.

Ishan Chatterjee, Robert Xiao, and Chris Harrison. Gaze+gesture: Expressive, precise and targeted free-space interactions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 131–138, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450339124. doi: 10.1145/2818346.2820752. URL https://doi.org/10.1145/2818346.2820752.

Di Laura Chen, Ravin Balakrishnan, and Tovi Grossman. Disambiguation techniques for freehand object manipulations in virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 285–292, 2020. doi: 10.1109/VR46266.2020.00048.

Eunjung Choi, Heejin Kim, and Min K. Chung. A taxonomy and notation method for three-dimensional hand gestures. *International Journal of Industrial Ergonomics*, 44(1):171–188, 2014. ISSN 0169-8141. doi: https://doi.org/10.1016/j.ergon.2013.10.011. URL https://www.sciencedirect.com/science/article/pii/S0169814113001285.

Shreya Chopra and Frank Maurer. Evaluating user preferences for augmented reality interactions with the internet of things. In *Proceedings of the 2020 International Conference on Advanced Visual Interfaces*, AVI '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375351. doi: 10.1145/3399715.3399716. URL https://doi.org/10.1145/3399715.3399716.

Christopher Clarke and Hans Gellersen. *Dynamic motion coupling of body movement for input control*. PhD thesis, Lancaster University, 2020. URL https://eprints.lancs.ac.uk/id/eprint/145217.

James J. Gibson. The theory of affordances. In *The Ecological Approach to Visual Perception*, pages 119–135. Houghton Mifflin, Boston, MA, 1979.

Yves Guiard, Renaud Blanch, and Michel Beaudouin-Lafon. Object pointing: a com-

plement to bitmap pointing in guis. In *Proceedings of Graphics Interface 2004*, GI '04, page 9–16, Waterloo, CAN, 2004. Canadian Human-Computer Communications Society. ISBN 1568812272.

Maxime Guillon, François Leitner, and Laurence Nigay. Investigating visual feedforward for target expansion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 2777–2786, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. doi: 10.1145/2702123.2702375. URL https://doi.org/10.1145/2702123.2702375.

Violet Yinuo Han, Tianyi Wang, Hyunsung Cho, Kashyap Todi, Ajoy Savio Fernandes, Andre Levi, Zheng Zhang, Tovi Grossman, Alexandra Ion, and Tanya R. Jonker. A dynamic bayesian network based framework for multimodal context-aware interactions. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25, page 54–69, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713064. doi: 10.1145/3708359.3712070. URL https://doi.org/10.1145/3708359.3712070.

Ivy Xiao He, Stefanie Tellex, and Jason Xinyu Liu. Legs-pomdp: Language and gesture-guided object search in partially observable environments. In *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, HRI '26, page 227–236. ACM, March 2026. doi: 10.1145/3757279.3785585. URL http://dx.doi.org/10.1145/3757279.3785585.

Masoumehsadat Hosseini, Tjado Ihmels, Ziqian Chen, Marion Koelle, Heiko Müller, and Susanne Boll. Towards a consensus gesture set: A survey of mid-air gestures in hci for maximized agreement across domains. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY,

USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581420. URL https://doi.org/10.1145/3544548.3581420.

Xiyun Hu, Dizhi Ma, Fengming He, Zhengzhe Zhu, Shao-Kang Hsia, Chenfei Zhu, Ziyi Liu, and Karthik Ramani. Gesprompt: Leveraging co-speech gestures to augment llm-based interaction in virtual reality, 2025. URL https://arxiv.org/abs/2505.05441.

Siddarth Jain and Brenna Argall. Probabilistic human intent recognition for shared autonomy in assistive robotics. *J. Hum.-Robot Interact.*, 9(1), December 2019. doi: 10.1145/3359614. URL https://doi.org/10.1145/3359614.

Ed Kaiser, Alex Olwal, David McGee, Hrvoje Benko, Andrea Corradini, Xiaoguang Li, Phil Cohen, and Steven Feiner. Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, page 12–19, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136218. doi: 10.1145/958432.958438. URL https://doi.org/10.1145/958432.958438.

Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. doi: 10.1145/3173574.3173655. URL https://doi.org/10.1145/3173574.3173655.

Jaewook Lee, Jun Wang, Elizabeth Brown, Liam Chu, Sebastian S. Rodriguez, and Jon E. Froehlich. Gazepointar: A context-aware multimodal voice assistant for pronoun disambiguation in wearable augmented reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*,

CHI '24, page 1–20. ACM, May 2024. doi: 10.1145/3613904.3642230. URL http://dx.doi.org/10.1145/3613904.3642230.

Yaxiong Lei, Shijing He, Mohamed Khamis, and Juan Ye. An end-to-end review of gaze estimation and its interactive applications on handheld mobile devices. *ACM Comput. Surv.*, 56(2), September 2023. ISSN 0360-0300. doi: 10.1145/3606947. URL https://doi.org/10.1145/3606947.

Zhuoming Li, Aitong Liu, Mengxi Jia, Yubo Lu, Tengxiang Zhang, Changzhi Sun, Dell Zhang, and Xuelong Li. Gestura: A lvlm-powered system bridging motion and semantics for real-time free-form gesture understanding. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(4):1–29, December 2025. ISSN 2474-9567. doi: 10.1145/3770709. URL http://dx.doi.org/10.1145/3770709.

Tica Lin, Ben Lafreniere, Yan Xu, Tovi Grossman, Daniel Wigdor, and Michael Glueck. Xr input error mediation for hand-based input: Task and context influences a user's preference, 2023. URL https://arxiv.org/abs/2309.10899.

Diako Mardanbegi, Tobias Langlotz, and Hans Gellersen. Resolving target ambiguity in 3d gaze interaction through vor depth estimation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300842. URL https://doi.org/10.1145/3290605.3300842.

Sharon Oviatt. Mutual disambiguation of recognition errors in a multimodel architecture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, page 576–583, New York, NY, USA, 1999. Association

for Computing Machinery. ISBN 0201485591. doi: 10.1145/302979.303163. URL https://doi.org/10.1145/302979.303163.

Jeongeun Park, Seungwon Lim, Joonhyung Lee, Sangbeom Park, Minsuk Chang, Youngjae Yu, and Sungjoon Choi. Clara: Classifying and disambiguating user commands for reliable interactive robotic agents, 2024. URL https://arxiv.org/abs/2306.10376.

Thammathip Piumsomboon, David Altimira, Hyungon Kim, Adrian Clark, Gun Lee, and Mark Billinghurst. Grasp-shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 73–82, 2014. doi: 10.1109/ISMAR.2014.6948411.

Siddharth Reddy, Sergey Levine, and Anca D. Dragan. First contact: Unsupervised human-machine co-adaptation via mutual information maximization, 2022. URL https://arxiv.org/abs/2205.12381.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners, 2023. URL https://arxiv.org/abs/2307.01928.

Julia Schwarz, Jennifer Mankoff, and Scott Hudson. Monte carlo methods for managing interactive state, action and feedback under uncertainty. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 235–244, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450307161. doi: 10.1145/2047196.2047227. URL https://doi.org/10.1145/2047196.2047227.

Julia Schwarz, Jennifer Mankoff, and Scott E. Hudson. An architecture for generating interactive feedback in probabilistic user interfaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 2545–2554, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. doi: 10.1145/2702123.2702228. URL https://doi.org/10.1145/2702123.2702228.

Soroush Shahi, Vimal Mollyn, Cori Tymoszek Park, Runchang Kang, Asaf Liberman, Oron Levy, Jun Gong, Abdelkareem Bedri, and Gierad Laput. Vision-based hand gesture customization from a single demonstration. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, page 1–14. ACM, October 2024. doi: 10.1145/3654777.3676378. URL http://dx.doi.org/10.1145/3654777.3676378.

Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376438. URL https://doi.org/10.1145/3313831.3376438.

Ludwig Sidenmark, Mark Parent, Chi-Hao Wu, Joannes Chan, Michael Glueck, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. Weighted pointer: Error-aware gaze-based interaction through fallback modalities. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3585–3595, 2022. doi: 10.1109/TVCG.2022.3203096.

Donald Tanguay. Hidden markov models for gesture recognition. 1995. URL https://api.semanticscholar.org/CorpusID:12928540.

Ching-Yi Tsai, Nicole Tacconi, Andrew D. Wilson, and Parastoo Abtahi. Uncertain pointer: Situated feedforward visualizations for ambiguity-aware ar target selection. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722783. doi: 10.1145/3772318.3790329. URL https://doi.org/10.1145/3772318.3790329.

Eduardo Velloso and Carlos H Morimoto. A probabilistic interpretation of motion correlation selection techniques. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, page 1–13. ACM, May 2021. doi: 10.1145/3411764.3445184. URL http://dx.doi.org/10.1145/3411764.3445184.

Zhimin Wang, Haofei Wang, Huangyue Yu, and Feng Lu. Interaction with gaze, gesture, and speech in a flexibly configurable augmented reality system. *IEEE Transactions on Human-Machine Systems*, 51(5):524–534, 2021. doi: 10.1109/THMS.2021.3097973.

Adam S. Williams, Jason Garcia, and Francisco Ortega. Understanding multimodal user gesture and speech behavior for object manipulation in augmented reality using elicitation. *IEEE Transactions on Visualization and Computer Graphics*, 26 (12):3479–3489, 2020. doi: 10.1109/TVCG.2020.3023566.

John Williamson and Roderick Murray-Smith. Pointing without a pointer. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, page 1407–1410, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581137036. doi: 10.1145/985921.986076. URL https://doi.org/10.1145/985921.986076.

Andy Wilson and Steven Shafer. Xwand: Ui for intelligent spaces. In *CHI '03 Proceedings of the SIGCHI Conference on Human Fac-*

tors in Computing Systems, pages 545–552. ACM, April 2003. URL https://www.microsoft.com/en-us/research/publication/xwand-ui-intelligent-spaces/.

Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 1083–1092, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605582467. doi: 10.1145/1518701.1518866. URL https://doi.org/10.1145/1518701.1518866.

Siju Wu, Amine Chellali, Samir Otmane, and Guillaume Moreau. Horizontaldragger: A freehand remote selector for object acquisition. In *2015 IEEE Virtual Reality (VR)*, pages 307–308, 2015. doi: 10.1109/VR.2015.7223418.

Haijun Xia, Michael Glueck, Michelle Annett, Michael Wang, and Daniel Wigdor. Iteratively designing gesture vocabularies: A survey and analysis of best practices in the hci literature. *ACM Trans. Comput.-Hum. Interact.*, 29(4), May 2022. ISSN 1073-0516. doi: 10.1145/3503537. URL https://doi.org/10.1145/3503537.

Pengfei Xu, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-Lan Tai. Lazy selection: a scribble-based tool for smart shape elements selection. *ACM Trans. Graph.*, 31(6), November 2012. ISSN 0730-0301. doi: 10.1145/2366145.2366161. URL https://doi.org/10.1145/2366145.2366161.

Xin Zeng, Xiaoyu Wang, Tengxiang Zhang, Chun Yu, Shengdong Zhao, and Yiqiang Chen. Gesturegpt: Toward zero-shot free-form hand gesture understanding with large language model agents. *Proceedings of the ACM on Human-Computer Interaction*, 8(ISS):462–499, October 2024. ISSN 2573-0142. doi: 10.1145/3698145. URL http://dx.doi.org/10.1145/3698145.

Xiaoyan Zhou, Adam Sinclair Williams, and Francisco Raul Ortega. Eliciting multimodal gesture+speech interactions in a multi-object augmented reality environ-

ment. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*, VRST '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450398893. doi: 10.1145/3562939.3565637. URL https://doi.org/10.1145/3562939.3565637.