OMNISMARTHOME: A MULTIMODAL REASONING BENCHMARK FOR SMART-HOME AGENTS

Anonymous authors
Paper under double-blind review

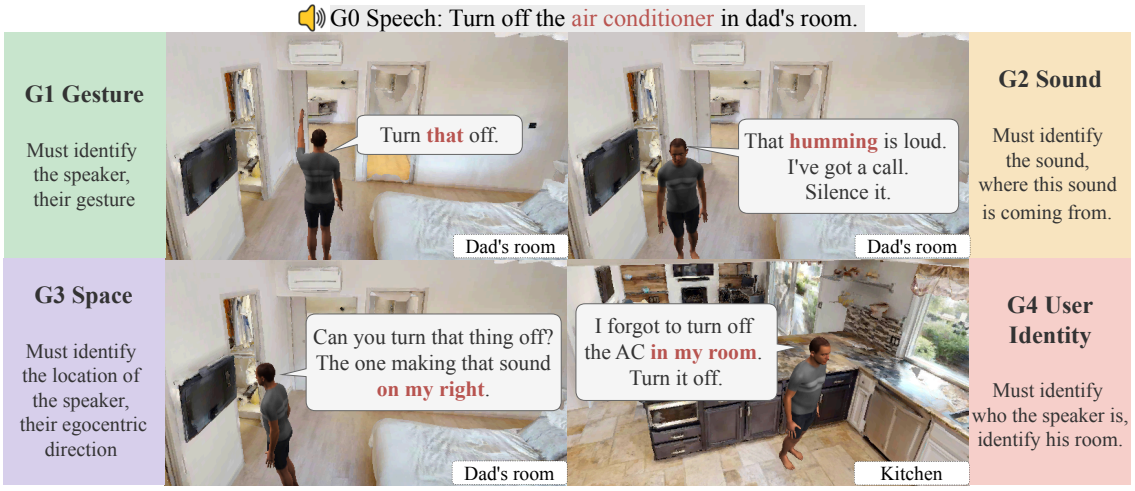


Figure 1: **Overview of OmniSmartHome.** In real-world interactions, a user's spoken request is often underspecified in language alone, with the intended target device resolved from multimodal cues distributed across the surrounding audio-visual context (G1-G4).

ABSTRACT

Smart-home assistants are expected to handle diverse, realistic requests that arise in daily life. In such interactions, users often rely on the surrounding multimodal context—pointing at objects or referring to what they see or hear, leaving their requests underspecified in language alone. Existing smart-home benchmarks, however, express user requests solely through language, leaving context-dependent real-world requests underexplored. To bridge this gap, we introduce OmniSmartHome, a multimodal smart-home benchmark where each spoken request is paired with the surrounding visual and spatial-audio context, providing complementary cues to disambiguate underspecified requests. OmniSmartHome comprises 1,360 synthetic and 272 real-world episodes. We evaluate 16 omnimodal large language models (Omni-LLMs) and reveal that, while they perform strongly when speech alone sufficiently conveys the user's intent, performance drops substantially when resolving it requires reasoning over multimodal contextual cues. As a simple agent baseline, we provide PROME (PROcedural Memory for multimodal Evidence gathering), which equips agents with specialized audio-visual perception tools and procedural memory for orchestrating their use. PROME generally improves performance across six Omni-LLMs. Demos and examples are available at `https://omni-smart-home.github.io`.

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

1 INTRODUCTION

Smart-home assistants control home devices on a user's behalf, translating each request into a device operation. Commercial systems such as Amazon Alexa have made voice-controlled home device operation widely accessible, yet they remain largely centered on explicit verbal commands. In everyday interaction, however, people often make requests that are ambiguous from language alone, as they rely on the surrounding multimodal context while speaking—pointing at objects or referring to what they see or hear (Tomasello, 2008; Clark, 1996). Interpreting such requests therefore depends on that context: resolving "*Turn that off*" requires the accompanying gesture, while resolving "*Silence that humming noise*" requires identifying the source of the sound. Supporting such context-grounded requests is particularly beneficial for users with impaired memory or limited expressive ability, such as young children and individuals with cognitive or communication impairments (So et al., 2010; Hendriks et al., 2014; Jiang et al., 2026). Next-generation smart-home agents powered by large language models (LLMs) are thus expected to handle such everyday requests with their multimodal capabilities.

Benchmarks for LLM-based smart-home agents have increasingly incorporated the complexities of real-world interaction, covering invalid user requests (Li et al., 2025), dynamically changing environments (Seo et al., 2026), and personalized dialogue (Li et al., 2026b; Bharadwaj et al., 2026). These benchmarks, while capturing the diverse and dynamic nature of everyday interactions, are composed of requests conveyed through language alone. Despite the importance of surrounding multimodal context in everyday communication, no existing smart-home benchmark evaluates agents on requests grounded in such context.

To address this gap, we introduce **OmniSmartHome**, a multimodal smart-home benchmark in which each spoken request is accompanied by the surrounding visual and spatial-audio context, which provides complementary cues for disambiguating the underspecified request. OmniSmartHome categorizes requests into five types by how the target device is indicated in user's request: *speech*, where it is named explicitly, and *gesture*, *sound*, *space*, and *user identity*, where speech alone is ambiguous and the target is inferred from the corresponding multimodal contextual cues. As illustrated in Fig. 1, a target explicitly named in *speech* as "*air conditioner in dad's room*" may instead be referred to as "*that*" with a pointing gesture (*gesture*), by its sound ("*humming*"; *sound*), by its relative location ("*on my right*"; *space*), or through a possessive reference ("*my room*"; *user identity*). Fulfilling such requests involves two steps: identifying the target device and executing the requested operation on it. We evaluate these steps with *grounding* and *goal* accuracy, respectively.

The grounding step is the new challenge introduced by OmniSmartHome: it requires *perceiving* fine-grained cues from audio-visual context, and *aggregating* them across modalities. Recent omnimodal large language models (Omni-LLMs), despite accepting audio-visual input, struggle to perceive such cues as pointing direction, sound source location, spatial relations, and speaker identity (Choi et al., 2026; Li et al., 2026a; Chen et al., 2025; Bai et al., 2026), and even when the cues are accessible, often fail to integrate them into a coherent decision (Luong et al., 2026; LI et al., 2025; Zhang et al., 2026a). Evaluating 16 Omni-LLMs on OmniSmartHome exposes these limitations: models perform well when the target device is explicitly named in *speech*, but struggle to infer it from multimodal contextual cues, leaving a clear gap to human performance. As a simple agent baseline, we provide PROME (PROcedural Memory for multimodal Evidence gathering), which combines multimodal tools for fine-grained perception with procedural memory for orchestrating them. Across six Omni-LLMs, PROME improves both grounding and goal accuracy.

Our contributions are threefold: (1) We introduce OmniSmartHome, the first multimodal smart-home benchmark in which spoken requests are accompanied by surrounding visual and spatial-audio context that provides complementary cues—gesture, sound, space, user identity—for resolving ambiguous requests. (2) We evaluate 16 Omni-LLMs, showing that current models struggle to interpret user requests that depend on multimodal context, while performing strongly when the request is fully specified in speech alone. (3) To narrow this gap, we propose PROME, a simple agent baseline that pairs specialized audio-visual perception tools with a procedural memory, improving both grounding and goal accuracy across six Omni-LLMs.

Table 1: **Comparison with related smart-home benchmarks.** OmniSmartHome covers diverse modalities spanning text, visual, and audio, and provides both synthetic and real-world data.

Benchmark	Modality				Data Type		Data Size
	Text	Visual	Speech Instruction	General Audio	Synthetic	Real-world	
HomeBench (Li et al., 2025)	✓	✗	✗	✗	✓	✗	170k
SimuHome (Seo et al., 2026)	✓	✗	✗	✗	✓	✗	0.6k
PersonalHomeBench (Bharadwaj et al., 2026)	✓	✓	✗	✗	✓	✓	9k
SMH-Bench (Li et al., 2026b)	✓	✗	✗	✗	✓	✗	1.1k
MIST (Chen et al., 2026)	✓	✗	✓	✗	✓	✗	10k
OmniSmartHome	✓	✓	✓	✓	✓	✓	**1.6k**

2 Related Works

Smart-home agents. Smart-home agents fulfill user requests by operating home devices through tool calls. A growing number of benchmarks have been proposed to evaluate such agents under realistic interactions (Table 1). HomeBench (Li et al., 2025) includes both valid and invalid requests. SimuHome (Seo et al., 2026) introduces an interactive simulator in which the agent's device-control calls update device states and, in turn, environmental variables. It further introduces a simulator-based evaluation protocol that evaluates agents by the resulting simulator state. SMH-Bench (Li et al., 2026b) incorporates ambiguous instructions, multi-turn interaction, and personalization. More recent benchmarks have begun to incorporate non-textual inputs, but only partially. PersonalHomeBench (Bharadwaj et al., 2026) includes silent video, but primarily evaluates video understanding rather than device control, while MIST (Chen et al., 2026) replaces text with speech but does not require broader multimodal context, such as general audio or visual cues, to resolve the request. In contrast, OmniSmartHome incorporates spatial audio containing both speech and general environmental sounds, alongside visual observations, requiring agents to reason over such multimodal context.

Omni-LLMs. Extending the capabilities of LLMs, Omni-LLMs have recently emerged, broadening text-centric understanding to encompass audio and visual perception. Omni-LLMs range from smaller-scale models (Tang et al., 2025; Xu et al., 2025a; Cui et al., 2026) to larger-scale models with strong reasoning capabilities (Xu et al., 2025b; Deshmukh et al., 2026; Team et al., 2026). Despite advances in holistic multimodal understanding, these models remain weak at perceiving fine-grained cues (Choi et al., 2026; Li et al., 2026a; Chen et al., 2025; Zheng et al., 2025; Bai et al., 2026), and, even when such cues are available, often fail to aggregate them into a coherent conclusion (Luong et al., 2026; LI et al., 2025; Xu et al., 2026; Zhang et al., 2026a). A growing line of work therefore augments multimodal LLMs with specialized perception tools (Yang et al., 2023; Chen et al., 2025) and learns to orchestrate them, typically with reinforcement learning (Tong et al., 2026; Zheng et al., 2026) or procedural memory (Wang & Chen, 2025; Si et al., 2026). Similarly, our agent baseline, PROME, defines a set of perception tools tailored to smart-home and builds a procedural memory to guide tool usage and evidence aggregation.

3 OmniSmartHome

We present the benchmark taxonomy (Sec. 3.1), dataset generation pipeline (Sec. 3.2), and evaluation protocol (Sec. 3.3), followed by a simple agent baseline (Sec. 3.4).

3.1 Benchmark Taxonomy

OmniSmartHome defines an *episode profile* along three dimensions: a *grounding type*, indicating which multimodal cue the user relies on to specify the target device; a *query type*, indicating what the user asks for; and a *feasibility type*, indicating whether the request can be fulfilled or not.

Grounding types. We define five *grounding types*: one explicit baseline (G0) and four implicit types (G1–G4) that rely on multimodal contextual cues. *Speech-grounded (G0).* The user explicitly names the target device in the spoken utterance, so grounding reduces to speech recognition alone. *Gesture-grounded (G1).* The user refers to the target device deictically (e.g., "Turn *that* off") while pointing toward it, requiring the agent to interpret the gesture and ground it in the visual scene. *Sound-grounded (G2).* The user refers to the target device by the sound it emits (e.g., "Silence that *humming noise*"), requiring the agent to detect and localize the sound source and associate it with the corresponding device. *Space-grounded (G3).* The user refers to the target device using an egocentric direction (e.g., "Turn off the light *on my right*" or "Silence that sound coming *from my right*"), requiring the agent to reason about the referenced direction from the user's perspective and identify the target located there. *User-Identity-grounded (G4).* The user refers to the target device through a possessive reference (e.g., "Turn off the AC in *my room*"), requiring the agent to identify the speaker by face or voice and resolve the reference against that person's household profile.

Query types. We define four query types, following SimuHome (Seo et al., 2026). *State inquiry (Q1).* The user asks about the current state or attributes of a device (e.g., "What temperature is the AC set to?"), requiring the agent to retrieve the relevant information and report it in a textual response. *Explicit device control (Q2).* The user asks the agent to set a specific device to a desired state (e.g., "Turn that off"), requiring the agent to execute the corresponding device operation. *Workflow scheduling (Q3).* The user requests one or more device actions to be executed at a future time or in coordination with another event (e.g., "In ten minutes, turn that off"), requiring the agent to schedule the appropriate workflow. *Implicit intent (Q4).* The user expresses a desired outcome without explicitly specifying the action to take (e.g., "it's too cold in my room"), requiring the agent to infer the underlying intent and determine the appropriate device operation. Since implicit-intent requests are difficult to combine with G1-G3 grounding types, Q4 is instantiated only for G0 and G4 episodes.

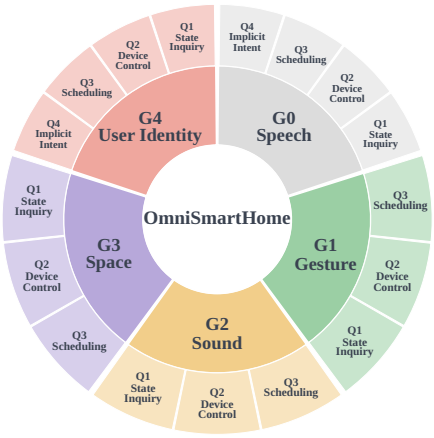


Figure 2: **Taxonomy of OmniSmartHome.** Each episode has a grounding, query, and feasibility type.

Feasibility types. We construct both *feasible* and *infeasible* episodes. For the synthetic dataset, feasible episodes are further divided into three difficulty levels—*easy*, *medium*, and *hard*—based on the distractors in the scene. For example, in gesture-grounded (G1) episodes, easy scenes feature a single candidate device; medium scenes contain multiple devices requiring precise gesture grounding; and hard scenes introduce non-speakers, necessitating active speaker detection prior to gesture recognition. For infeasible episodes, we consider three causes of infeasibility: *non-existent resources*, where the referenced device or perceptual cue is absent; *physical limits*, where the relevant device exists but cannot realize the requested change due to its operational limits; and *temporal contradictions*, where the requested timing conflicts with other time constraints or device operating conditions. (Details in Appendix A.)

3.2 Dataset Generation

Definition and components of an episode. Each episode comprises four components. (1) The *visual and acoustic observations* capture the room where the user is currently located: the visual observation shows the room and any people present, and the acoustic observation is a first-order Ambisonics (FOA) recording of the user's spoken request along with sounds from active devices. (2) The *initial home state* defines the starting configuration of the home: the devices in each room, their positions and initial states, and household-member profiles. (3) The *target device* is the ground-truth device that the request intends to control or query. (4) The *goal* specifies the correct outcome: the ground-truth state or attribute of the target device for Q1, and the desired state of the target device for Q2–Q4.

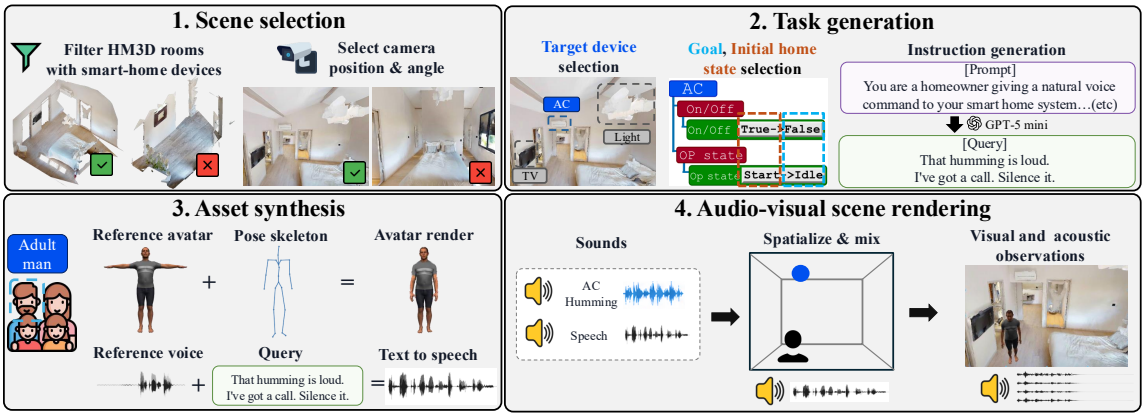


Figure 3: **Synthetic episode generation pipeline.** (1) Select an room containing suitable smart-home devices and a camera viewpoint. (2) Determine the target device, goal and initial home state and generate a corresponding user request in text. (3) Synthesize the avatar and speech, prepare device sounds. (4) Place the assets in the scene and spatialize the audio sources to produce the final visual and acoustic observations.

Synthetic episodes. We construct synthetic episodes in four stages, as illustrated in Fig. 3. (1) *Scene selection.* Scenes are drawn from the test split of HM3D (Yadav et al., 2023), a collection of photorealistic 3D meshes of indoor home environments. We retain only rooms containing suitable home devices and manually select camera viewpoints that provide clear visibility for each room. (2) *Task generation.* For each room, we assign episode profiles (grounding, query, and feasibility types). Conditioned on this profile, we select an appropriate *target device* and *goal*, and construct the *initial home state* such that the goal is initially unsatisfied. An LLM (Singh et al., 2025) then generates a natural user request in text consistent with the profile and goal. (3) *Asset synthesis.* We sample one of four demographic categories (girl, boy, adult woman, adult man), render a matching humanoid avatar (Puig et al., 2024) in a natural pose, and synthesize the user request as speech with a matching voice using recent speech synthesis model (Zhang et al., 2026b). For episodes requiring device sounds (e.g., humming, media playback, beeping, or chimes), we use a device-specific audio clip collected from Freesound (Font et al., 2013). (4) *Audio-visual scene rendering.* Using the 3D audio-visual simulator SoundSpaces 2.0 (Chen et al., 2022a), we place the selected avatar in the scene and emit the synthesized speech from the avatar's position, along with device-specific sounds from the corresponding device locations. This yields the *visual and acoustic observation* (a rendered image and an FOA audio).

Real-world episodes. We construct real-world episodes in five indoor environments. Five participants act out LLM-generated queries (Singh et al., 2025), which we record using a camera and a spatial-audio microphone. Unlike the synthetic set with static scene images, the real-world set consists of video recordings.

All episodes are manually verified by human (Appendix F.1). We retain, for each grounding–query combination, 20 synthetic episodes per feasible difficulty level and 20 infeasible episodes, and eight feasible and eight infeasible real-world episodes, yielding 1,360 synthetic and 272 real-world episodes.

3.3 Episode Execution and Evaluation

Episode execution. Fig. 4, excluding the orange components, illustrates the full architecture of the base agent. Each episode begins by initializing the simulator (Seo et al., 2026) with the *initial home state*, while the Omni-LLM agent receives the *visual and acoustic observations*. The agent operates in a ReAct loop (Yao et al., 2023), alternating between reasoning, tool invocation, and observation. At each step, the agent can invoke simulator-interaction tools as well as two dedicated tools, *Device Declare* and *Finish*. Simulator-interaction tools access the simulator either to retrieve information about the device state or to modify it by executing

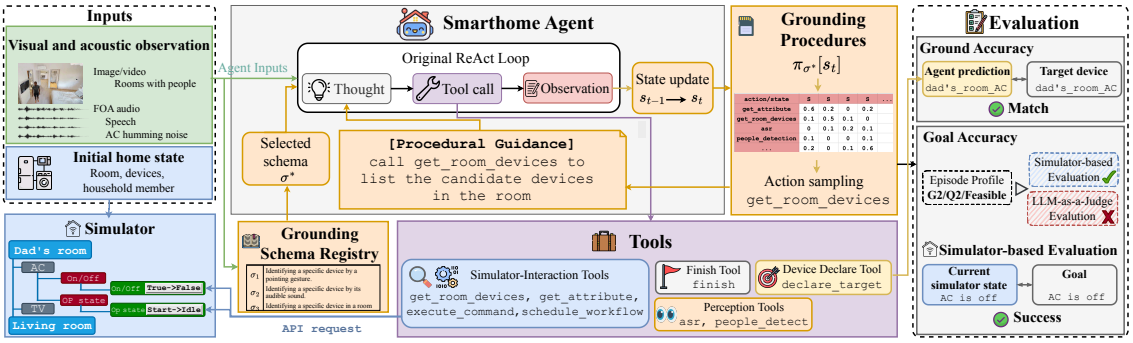


Figure 4: **Illustration of our agent framework.** Components highlighted in orange are used only by PROME. The agent receives the visual and acoustic observations and iteratively gathers evidence to identify the target device using simulator-interaction or perception tools. PROME additionally provides procedural guidance. Once the target device is identified, the agent declares it and either queries its information (Q1) or executes the requested operation (Q2-Q4) via simulator-interaction tools. Grounding accuracy evaluates the declared target device, while goal accuracy is evaluated from the simulator state or by an LLM judge.

device operations. The *Device Declare* tool records the agent's predicted target device and is required before the *Finish* tool can be called, which terminates the episode and returns the final textual response.

Evaluation metrics. We evaluate agents using two metrics: *grounding accuracy* and *goal accuracy*. *Grounding accuracy* measures whether the device declared via the *Device Declare* tool matches the ground-truth target device. For infeasible requests due to *non-existent resources*, where no valid target exists, the agent is expected to declare no target. *Goal accuracy* measures whether the agent ultimately satisfies the requested goal, evaluated using either a simulator or an LLM-as-a-judge (Zheng et al., 2023; Liu et al., 2023). For feasible Q2–Q4 queries, we inspect the final simulator state and check whether it matches the state specified in the ground-truth goal. For Q1 queries and infeasible requests, where the agent is expected to return a textual response rather than execute a device operation, we use an LLM-as-a-judge. For Q1, the judge assesses whether the response is consistent with the states and attributes of the target device; for infeasible requests, it assesses whether the agent correctly recognizes the infeasibility and explains its cause.

3.4 A Simple Agent Baseline

Before operating a device, an agent first resolves which device the user refers to from the audio-visual context. This involves actively gathering fine-grained multimodal cues and integrating them differently depending on how the request is grounded in the environment—recent Omni-LLMs often struggle with (Choi et al., 2026; Li et al., 2026a; Chen et al., 2025; Bai et al., 2026; Luong et al., 2026; LI et al., 2025). To support this process, we introduce **PROME** (**PRO**cedural **M**emory for multimodal **E**vidence gathering) as a simple agent baseline. PROME equips the agent with a set of audio-visual perception tools and augments its reasoning with a procedural memory that guides when such tools should be invoked and in what order.

Multimodal perception tools. We provide the agent with perceptual tools including speech recognition, pointing estimation, sound localization, head pose recognition, face recognition and speaker recognition (Radford et al., 2023; Shimada et al., 2025; Jiang et al., 2024; Zhou et al., 2023; He et al., 2017; Oquab et al., 2024; Chen et al., 2022b). Further details are provided in Appendix B.

Procedural memory. Procedural memory guides the agent by suggesting the next tool-calling action, based on which contextual cue the request relies on to indicate the target device and the evidence gathered so far. It consists of a bank of reusable *grounding procedures*, $\mathcal{M} = \{\pi_\sigma\}_{\sigma \in \Sigma}$, where each π_σ encodes the tool-calling strategy for a *grounding schema* $\sigma \in \Sigma$, characterized by the type of contextual cue that identifies the target device (e.g., a pointing gesture resolving "this" or "that"). Formally, π_σ defines a probability distribution

over tool actions $a \in \mathcal{A}$ conditioned on the current agent state $s_t \in \mathcal{S}$, which summarizes the evidence acquired and the remaining uncertainty at ReAct step t. We parameterize π_σ as a tabular softmax policy, $\pi_\sigma(a \mid s) \propto \exp(\theta_{\sigma,s,a})$, with learnable logits $\theta \in \mathbb{R}^{|\Sigma|\times|\mathcal{S}|\times|\mathcal{A}|}$.

At inference time, the agent first examines the multimodal input and retrieves the grounding schema $\sigma^* \in \Sigma$ that best matches the episode. After each ReAct step, the agent updates the current state s_t based on the evidence accumulated so far, samples $a_t \sim \pi_{\sigma^*}[s_t]$, and injects the sampled action as a brief [PROCEDURAL GUIDANCE] before the next reasoning step.

Procedural memory construction. We build procedural memory from synthetic training episode, generated using the same pipeline as in Sec. 3.2 with scenes drawn from the HM3D training split. To construct Σ, we prompt an Omni-LLM (Xu et al., 2025b) to describe, for each training episode, which cue is needed to identify the target device, and to group episodes with similar cue descriptions into schemas. For π_σ, we initialize the logits θ^0 as a cost prior plus a utility bonus for each perception tool, where the utility is estimated from two memory-free reference rollouts per episode: τ^+_{ref} with perception tools and τ^-_{ref} without. We then refine θ^0 with a policy-gradient update (Williams, 1992):

$$\nabla_\theta \mathcal{J} = \mathbb{E}_e \, \mathbb{E}_{\tau\sim\mathcal{G}_e}\Big[A(\tau) \sum_{t\in\mathcal{T}(\tau)} \nabla_\theta \log \pi_{\sigma^*}[s_t](a_t)\Big], \qquad A(\tau) = G(\tau) - \tfrac{1}{|\mathcal{G}_e|}\textstyle\sum_{\tau'\in\mathcal{G}_e} G(\tau'), \tag{1}$$

where $\mathcal{G}_e = \{\tau, \tau^+_{\text{ref}}, \tau^-_{\text{ref}}\}$ adds a memory-guided rollout τ sampled under θ^0 to the two reference rollouts, $G(\tau)$ is the grounding reward, and $\mathcal{T}(\tau)$ is the set of decision steps. The training set is rolled out once and updated in a single batch (details in Appendix D.1).

4 EXPERIMENTS

We evaluate 16 Omni-LLMs: seven open-source models with fewer than 10B parameters, five larger open-source models, and four proprietary models. For models that do not support spatial audio input, we downmix the FOA recordings to mono. To construct PROME memory, we use 680 synthetic training episodes from the HM3D training split, generated using the pipeline described in Sec. 3.2, and apply PROME to six models.

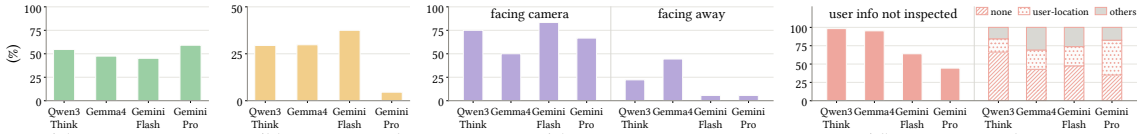
4.1 BENCHMARKING OMNI-LLMS ON OMNISMARTHOME

Main results. Tab. 2 reports the results on OmniSmartHome with base agents. Small open-source models show generally low grounding accuracy across all grounding types, with MiniCPM-o-4.5 performing best in this group. Their goal accuracy is often substantially lower than their grounding accuracy, in some cases by more than a factor of two, indicating that even when these models identify the target device, they still struggle to complete the requested goal. Larger open-source models achieve higher overall accuracy, with a smaller gap between grounding and goal performance. Their grounding performance is strongest on G0, where identifying the target device requires little beyond speech recognition, but drops on G1–G4, which require integrating additional multimodal evidence. Proprietary models generally outperform open-source models, yet show a similar gap between G0 and G1–G4. For these stronger models, the contrast between nearly saturated grounding and goal performance on G0 and lower performance on G1–G4 highlights multimodal grounding as the primary challenge of OmniSmartHome. Additional analysis is provided in Appendix C.

Error analysis. Fig. 5 examines four representative grounding failures across Qwen3-Omni-Think, Gemma4, Gemini-2.5-Flash, and Gemini-2.5-Pro. **G1.** In around 50% of the episodes where the agent declares an incorrect device, the chosen device is the one nearest to the speaker's hand or body (Fig. 5.a). This suggests that agents often fail to trace the pointing direction and instead fall back to selecting a device near the gesture itself. **G2.** Fig. 5.b reports, among the episodes where tbe agent declares an incorrect device, the fraction in which the agent declares the sound source prematurely, i.e., without checking the state of any candidate

Table 2: **Results on OmniSmartHome across grounding types (G0–G4).** For each grounding type, we report grounding accuracy (Gr.) and goal accuracy (Goal), averaged over query types.

	All		**G0** Speech		**G1** Gesture		**G2** Sound		**G3** Space		**G4** Identity	
Model	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal
Human Level	94.8	82.3	97.8	82.1	97.3	89.4	93.1	91.7	93.4	84.1	91.7	80.4
Open-source Omni-LLMs (<10B)												
Video-LLaMA2 7B	5.8	4.9	6.8	3.4	6.2	6.9	6.2	6.2	7.6	6.2	2.9	2.9
Qwen2.5-Omni 7B	33.0	17.6	55.5	25.5	30.9	19.4	28.1	14.2	27.2	17.4	20.3	10.9
Spatial-Omni 7B	20.0	11.5	39.1	14.1	14.9	11.5	16.3	10.4	16.0	11.8	10.4	9.4
video-SALMONN2+ 7B	27.0	13.7	48.2	20.6	20.1	11.8	23.6	13.5	24.7	14.9	15.4	7.6
video-SALMONN-o1 7B	24.5	10.2	44.5	13.5	24.7	10.4	17.4	9.4	25.3	11.8	9.1	6.5
OmniVinci 9B	17.8	8.8	31.2	9.9	9.0	6.9	14.9	9.7	17.0	9.7	13.5	7.6
MiniCPM-o-4.5 9B	60.6	30.9	85.2	51.0	56.9	29.9	56.6	26.4	59.4	32.3	42.7	14.1
Large Open-source Omni-LLMs												
Gemma4 12B	50.6	25.2	69.3	45.3	46.5	19.1	55.2	29.2	54.9	26.0	28.1	6.0
Nemotron3-Omni 30B-A3B	62.7	44.5	86.5	60.9	65.3	52.4	67.4	47.2	64.9	51.4	31.8	15.1
Qwen3-Omni-Instruct 30B-A3B	66.6	43.8	88.0	63.0	66.3	53.8	77.4	38.2	60.8	48.6	41.7	17.4
Qwen3-Omni-Think 30B-A3B	65.9	48.4	87.8	67.4	66.0	56.6	79.9	56.9	72.2	55.2	28.6	11.7
MiMo-V2.5 310B-A15B	78.4	60.2	93.5	73.7	72.2	63.5	81.9	60.8	84.7	69.4	60.7	37.0
Proprietary Omni-LLMs												
Gemini-2.5-Flash	73.7	56.2	94.0	76.8	69.4	60.1	74.3	54.9	78.1	63.2	52.6	28.6
Gemini-2.5-Pro	81.0	63.1	95.1	80.5	74.7	66.7	81.9	50.3	83.0	69.8	69.5	47.7
Gemini-3.1-Pro (Preview)	85.6	73.4	95.3	81.0	76.4	71.5	97.6	80.9	87.5	79.2	72.4	57.3
Qwen3.8-Omni-Flash	81.7	63.1	93.2	76.8	76.7	64.2	92.7	66.0	85.4	72.2	63.0	39.6



(a) G1 Gesture (b) G2 Sound (c) G3 Space (d) G4 User Identity

Figure 5: **Representative causes of grounding failures of the base agents.** For each grounding type (G1–G4), we show a representative failure pattern.

device, such as whether it is running or its alarm is active. Instead, the agent often relies on semantic priors about which devices typically produce a given sound: for instance, an alarm from a refrigerator is attributed to a microwave, since the agent assumes that microwaves commonly emit alarm sounds. **G3.** Fig. 5.c shows failure rates by speaker orientation. When the speaker faces the camera, their left–right directions are reversed relative to the image, and three models fail on over 70% of such cases, whereas error rates remain low when the speaker faces away. This suggests that models struggle to interpret directions from the speaker's egocentric view. **G4.** The left of Fig. 5.d shows, among failed episodes, the fraction in which the agent does not inspect any user-related information. Rather than querying user information to identify the device associated with the speaker's room, most of these cases end up either declaring no valid device (hatched bars) or selecting a device in the room where the user is currently located (dotted bars), as shown on the right.

4.2 Results on PROME

As shown in Tab. 3a, PROME improves overall grounding and goal accuracy across six Omni-LLMs. Notably, Gemma4, the weakest larger open-source model, gains 23.4/34.1 percentage points in grounding/goal accuracy, respectively, surpassing the other open-source base agents. PROME also improves the strong Gemini models by 4.5/6.1 points on Gemini-2.5-Flash and 4.0/6.1 points on Gemini-2.5-Pro. Across grounding types, as expected, gains are limited on G0 but larger on G1–G4.

Table 3: **Results and analysis on PROME.** (a) Per-grounding-type results; small numbers denote the change in percentage points relative to the base agent. (b) Ablations on PROME. (c) Memory transfer across backbones. (d) Grounding accuracy on the real-world subset.

	All		**G0** Speech		**G1** Gesture		**G2** Sound		**G3** Space		**G4** Identity	
Model	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal
Qwen2.5-Omni + **PROME**	38.2 +5.2	22.4 +4.8	61.5 +6.0	34.6 +9.1	41.0 +10.1	28.1 +8.7	35.1 +7.0	14.2 +0.0	28.1 +0.9	22.9 +5.5	22.9 +2.6	11.7 +0.8
Gemma4 + **PROME**	74.0 +23.4	59.3 +34.1	86.2 +16.9	73.4 +28.1	65.3 +18.8	58.3 +39.2	89.6 +34.4	70.8 +41.6	75.7 +20.8	58.3 +32.3	55.5 +27.4	37.8 +31.8
Nemotron + **PROME**	65.2 +2.5	49.1 +4.6	88.8 +2.3	66.7 +5.8	64.9 -0.4	55.6 +3.2	70.1 +2.7	53.8 +6.6	70.8 +5.9	57.6 +6.2	33.9 +2.1	16.9 +1.8
Qwen3-Think + **PROME**	70.2 +4.3	53.9 +5.5	87.5 -0.3	69.8 +2.4	70.1 +4.1	58.3 +1.7	86.5 +6.6	68.8 +11.9	77.1 +4.9	62.2 +7.0	35.4 +6.8	17.2 +5.5
Gemini-2.5-Flash + **PROME**	78.2 +4.5	62.3 +6.1	91.9 -2.1	76.6 -0.2	72.2 +2.8	62.2 +2.1	81.2 +6.9	63.5 +8.6	77.8 -0.3	65.6 +2.4	67.2 +14.6	44.5 +15.9
Gemini-2.5-Pro + **PROME**	85.0 +4.0	69.2 +6.1	96.6 +1.5	79.4 -1.1	74.3 -0.4	66.3 -0.4	86.8 +4.9	68.4 +18.1	87.8 +4.8	73.6 +3.8	78.1 +8.6	58.3 +10.6

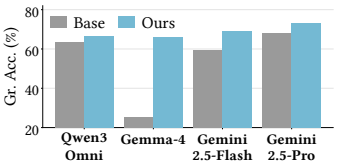
(a) Results across grounding types (G0–G4).

	Gemini-2.5-Flash		Gemini-2.5-Pro	
Method	Gr.	Goal	Gr.	Goal
Base	73.7	56.2	81.0	63.1
+Tool	75.2	58.9	83.9	65.9
+PROME	**78.2**	**62.3**	**85.0**	**69.2**

(b) Ablation of PROME.

	Gemma4		Nemotron3-Omni	
Method	Gr.	Goal	Gr.	Goal
Base	50.6	25.2	62.7	44.5
+Trans. Mem	71.7	51.7	64.9	48.7
+PROME	**74.0**	**59.3**	**65.2**	**49.1**

(c) Memory transfer.



(d) Real-world transfer.

Ablation studies on PROME. Tab. 3b isolates the effects of the perception tools and procedural memory on Gemini-2.5-Flash and Gemini-2.5-Pro. Perception tools alone provide modest gains on both models, while adding procedural memory yields further improvements.

Memory transfer. Tab. 3c applies procedural memory constructed with Qwen3-Omni-Think to Gemma4 and Nemotron3-Omni. Although each model performs best with its own memory, the transferred memory still improves over the base agent, showing that procedural memory transfers across models.

Real-world generalization. Tab. 3d reports grounding accuracy on the real-world subset. Although the procedural memory is learned entirely from synthetic data, PROME consistently improves grounding accuracy, suggesting that our synthetic episodes capture challenges that transfer to real-world interactions.

5 Conclusion

We introduce OmniSmartHome, a multimodal smart-home agent benchmark in which the information required to resolve a user request is distributed across modalities rather than explicitly stated in language. OmniSmartHome pairs each spoken request with visual and spatial-audio observations, covers both synthetic and real-world episodes, and spans five grounding types, four query types, and both feasible and infeasible requests. Our evaluation of 16 Omni-LLMs shows that current models reliably handle requests explicitly stated in speech but struggle when the target device is inferred from multimodal contextual cues. To narrow this gap, we propose PROME, a simple, model-agnostic agent baseline that combines specialized perception tools with a procedural memory guiding tool use. PROME consistently improves both grounding and goal accuracy. We envision OmniSmartHome as a step toward multimodal smart-home assistants that understand requests as naturally as people make them and, ultimately, act proactively on what they see and hear.

AI USE STATEMENT

In this work, we used generative AI tools as part of our benchmark evaluation procedure, such as the use of LLM-based judges, and to generate synthetic datasets. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results. Additionally, we used generative AI tools to improve the readability of the manuscript, draft parts of the paper, and assist in the development of a web-based interface used for human verification and human performance evaluation. We reviewed all AI-assisted work. In particular, AI-generated code and text were manually reviewed, verified, and revised by the authors. Human verification was also conducted for relevant evaluation procedures and artifacts. We take full responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Details on the models used in our experiments, including their providers and access methods, are provided in Appendix C.1. LLM prompts are in Appendix E. We also release the relevant implementation code in the supplementary material to facilitate reproducibility.

REFERENCES

Mimo-v2.5. https://huggingface.co/collections/XiaomiMiMo/mimo-v25, 2026.

Detao Bai, Shimin Yao, Weixuan Chen, Zhiheng Ma, Xihan Wei, and Jingren Zhou. Humanomni-speaker: Identifying who said what and when. *arXiv preprint arXiv:2603.21664*, 2026.

Manasa Bharadwaj, Yolanda Liu, InJung Yang, Sungil Kim, Nikhil Verma, KoKeun Kim, Kevin Ferreira, and YoungJoon Kim. Personalhomebench: Evaluating agents in personalized smart homes. *arXiv preprint arXiv:2604.16813*, 2026.

Changan Chen, Carl Schissler, Sanchit Garg, Philip Kobernik, Alexander Clegg, Paul Calamia, Dhruv Batra, Philip W Robinson, and Kristen Grauman. Soundspaces 2.0: A simulation platform for visual-acoustic learning. In *Proc. NeurIPS*, 2022a.

Maximillian Chen, Xuanming Zhang, Michael Peng, Zhou Yu, Alexandros Papangelis, and Yohan Jo. Mist: Multimodal interactive speech-based tool-calling conversational assistants for smart homes. *arXiv preprint arXiv:2605.06897*, 2026.

Mingfei Chen, Zijun Cui, Xiulong Liu, Jinlin Xiang, Caleb Zheng, Jingyuan Li, and Eli Shlizerman. SAVVY: Spatial awareness via audio-visual LLMs through seeing and hearing. In *Proc. NeurIPS*, 2025.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 2022b.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

Yura Choi, Roy Miles, Rolandos Alexandros Potamias, Ismail Elezi, Jiankang Deng, and Stefanos Zafeiriou. Do you see what i am pointing at? gesture-based egocentric video question answering. In *Proc. CVPR*, 2026.

Herbert H Clark. *Using language*. Cambridge university press, 1996.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

Amala Sanjay Deshmukh, Kateryna Chumachenko, Tuomas Rintamaki, Matthieu Le, Tyler Poon, Danial Mohseni Taheri, Ilia Karmanov, Guilin Liu, Jarno Seppanen, Arushi Goel, et al. Nemotron 3 nano omni: efficient and open multimodal intelligence. *arXiv preprint arXiv:2604.24954*, 2026.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory. In *Findings of ACL*, 2026.

Frederic Font, Gerard Roma, and Xavier Serra. Freesound technical demo. In *Proc. ACM MM*, 2013.

Gemini Team, Google DeepMind. Gemini 3.1 Pro model card. Technical report, Google DeepMind, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/. Accessed: 2026-9-12.

Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jimenez Gutierrez, Yiheng Shu, Chan Hee Song, Jiaman Wu, Shijie Chen, Hanane Nour Moussa, TIANSHU ZHANG, Jian Xie, Yifei Li, Tianci Xue, Zeyi Liao, Kai Zhang, Boyuan Zheng, Zhaowei Cai, Viktor Rozgic, Morteza Ziyadi, Huan Sun, and Yu Su. Mind2web 2: Evaluating agentic search with agent-as-a-judge. In *Proc. NeurIPS*, 2025.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. ICCV*, 2017.

Petra Hendriks, Charlotte Koster, and John CJ Hoeks. Referential choice across the lifespan: Why children and elderly adults produce ambiguous pronouns. *Language, cognition and neuroscience*, 29(4):391–407, 2014.

Chenxi Jiang, Chuhao Zhou, and Jianfei Yang. REI-bench: Can embodied agents understand vague human instructions in task planning? In *Proc. ICLR*, 2026.

Tao Jiang, Xinchen Xie, and Yining Li. Rtmw: Real-time multi-person 2d and 3d whole-body pose estimation. *arXiv preprint arXiv:2407.08634*, 2024.

Chentao Li, Zirui Gao, Mingze Gao, Yinglian Ren, Jianjiang Feng, and Jie Zhou. Do MLLMs understand pointing? benchmarking and enhancing referential reasoning in egocentric vision. In *Findings of ACL*, 2026a.

Kuan Li, Shuo Zhang, Huacan Wang, Fangzhou Yu, Zecheng Sheng, Yi Gu, Weipeng Ming, Lei Xue, Chen Liu, Sen Hu, et al. Smh-bench: Benchmarking llm agents for environment-grounded reasoning and action in smart homes. *arXiv preprint arXiv:2606.01912*, 2026b.

Silin Li, Yuhang Guo, Jiashu Yao, Zeming Liu, and Haifeng Wang. Homebench: Evaluating llms in smart homes with valid and invalid instructions across single and multiple devices. In *Proc. ACL*, 2025.

Yizhi LI, Yinghao Ma, Ge Zhang, Ruibin Yuan, King Zhu, Hangyu Guo, Yiming Liang, Jiaheng Liu, Zekun Moore Wang, Jian Yang, Siwei Wu, Xingwei Qu, Jinjie Shi, Xinyue Zhang, Zhenzhu Yang, Yidan WEN, Yanghai Wang, Shihao Li, Zhaoxiang Zhang, Ruibo Liu, Emmanouil Benetos, Wenhao Huang, and Chenghua Lin. Omnibench: Towards the future of universal omni-language models. In *Proc. NeurIPS*, 2025.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proc. EMNLP*, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *Proc. ICLR*, 2024.

Manh Luong, Tamas Abraham, Junae Kim, Amar Kaur, Rollin Omari, Gholamreza Haffari, Trang Vu, Lizhen Qu, and Dinh Phung. Mcbench: A multicontext safety assessment benchmark for omni large language models. *arXiv preprint arXiv:2606.05177*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *Proc. ICLR*, 2024.

Qwen Team. Qwen3.8-omni-flash: Omni senses. agentic delivery., September 2026. URL `https://qwen.ai/blog?id=qwen3.8-omni-flash`.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proc. ICML*, 2023.

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

Gyuhyeon Seo, Jungwoo Yang, Junseong Pyo, Nalim Kim, Jonggeun Lee, and Yohan Jo. Simuhome: A temporal- and environment-aware benchmark for smart home LLM agents. In *Proc. ICLR*, 2026.

Kazuki Shimada, Kengo Uchida, Yuichiro Koyama, Takashi Shibuya, Shusuke Takahashi, Yuki Mitsufuji, and Tatsuya Kawahara. Open-vocabulary sound event localization and detection with joint learning of clap embedding and activity-coupled cartesian doa vector. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 2025.

Shengyu Si, Yuanzhuo Lu, Ruimeng Yang, Ziyi Ye, Zuxuan Wu, and Yu-Gang Jiang. Vla-pro: Cross-task procedural memory transfer for vision-language-action models. *arXiv preprint arXiv:2605.29562*, 2026.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Wing Chee So, Özlem Ece Demir, and Susan Goldin-Meadow. When speech is ambiguous, gesture steps in: Sensitivity to discourse-pragmatic principles in early childhood. *Applied psycholinguistics*, 31(1):209–224, 2010.

Guangzhi Sun, Yudong Yang, Jimin Zhuang, Changli Tang, Yixuan Li, Wei Li, Zejun MA, and Chao Zhang. video-SALMONN-o1: Reasoning-enhanced audio-visual large language model. In *Proc. ICML*, 2025.

Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. video-salmonn 2: Caption-enhanced audio-visual large language models. *arXiv preprint arXiv:2506.15220*, 2025.

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

Michael Tomasello. *Origins of human communication*. The MIT Press, 2008.

Siqian Tong, Li Xuan, Yiwei Wang, Baolong Bi, Yujun Cai, Shenghua Liu, Yuchen He, and HAO Chengpeng. AuTAgent: A reinforcement learning framework for tool-augmented audio reasoning. In *Proc. ICML*, 2026.

Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*, 2025.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proc. ICML*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025a.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025b.

Ke Xu, Yuhao Wang, Ziyang Cheng, Hongcheng Liu, Yanfeng Wang, and Yu Wang. Agentic active omni-modal perception for multi-hop audio-visual reasoning. *arXiv preprint arXiv:2605.28192*, 2026.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. In *Proc. CVPR*, 2023.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. MM-REACT: Prompting ChatGPT for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *Proc. ICLR*, 2023.

Hanrong Ye, Chao-Han Huck Yang, Arushi Goel, Wei Huang, Zhen Wan, Jinchuan Tian, An-Chieh Cheng, Ligeng Zhu, Yuanhang Su, Yuming Lou, et al. Omnivinci: Enhancing architecture and data for omni-modal understanding llm. In *Proc. ICLR*, 2026.

Chenshuang Zhang, Kyeong Seon Kim, Chengxin Liu, and Tae-Hyun Oh. Svhalluc: Benchmarking speech-vision hallucination in audio-visual large language models. In *Proc. CVPR*, 2026a.

Yuqian Zhang, Donghua Yu, Zhengyuan Lin, Botian Jiang, Mingshu Chen, Yaozhou Jiang, Yiwei Zhao, Yiyang Zhang, Yucheng Yuan, Hanfu Chen, et al. Moss-ttsd: Text to spoken dialogue generation. *arXiv preprint arXiv:2603.19739*, 2026b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Proc. NeurIPS*, 2023.

Xu Zheng, Zihao Dongfang, Lutao Jiang, Boyuan Zheng, Yulong Guo, Zhenquan Zhang, Giuliano Albanese, Runyi Yang, Mengjiao Ma, Zixin Zhang, et al. Multimodal spatial reasoning in the large model era: A survey and benchmarks. *arXiv preprint arXiv:2510.25760*, 2025.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and XingYu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. In *Proc. ICLR*, 2026.

Huayi Zhou, Fei Jiang, and Hongtao Lu. Directmhp: Direct 2d multi-person head pose estimation with full-range angles. *arXiv preprint arXiv:2302.01110*, 2023.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *Proc. ICLR*, 2024.

Zhiyuan Zhu, Yixuan Chen, Yiwen Shao, Wenxiang Guo, Changhao Pan, Yu Zhang, Yuxiang Wang, Wei Liu, Houhua Zhang, Chengkuan Zeng, et al. Spatial-omni: Spatial audio understanding integration in multimodal llms via foa encoding. *arXiv preprint arXiv:2606.10738*, 2026.

Appendix

A Details on OmniSmartHome **18**

A.1 Audio-Visual Scene Synthesis 18

A.2 Details on Benchmark Design 18

A.3 Examples 19

A.4 Detail on Home Devices and Matter Cluster 20

B Details on Tools **24**

B.1 Original Tools 24

B.2 Perception Tools 24

C Additional Results on OmniSmartHome **25**

C.1 Baseline Models 25

C.2 Additional Results by Query Type 26

C.3 Additional Results by Difficulty Level 28

C.4 Modality Ablation 29

C.5 Error Analysis 29

D PROME **30**

D.1 Details of PROME 30

D.2 Real-world Transfer 32

D.3 Oracle Tool Calling 33

D.4 Comparison with other methods 33

D.5 Latency Analysis 34

E LLM Prompt **35**

E.1 Grounding Schema Registry Construction Prompt 35

E.2 Q1 Feasible Judge Prompt 35

E.3 Infeasible Judge Prompt 36

F Human Annotation **38**

F.1 Human Dataset Verifier 38

F.2 Human Performance 38

G Limitation **38**

A Details on OmniSmartHome

A.1 Audio-Visual Scene Synthesis

Speech and device sound sources. All spoken utterances are synthesized with MOSS-TTSD Zhang et al. (2026b), which natively outputs 24 kHz waveforms. Device sounds are drawn from a curated bank of 42 recordings covering every device–state pair referenced by the benchmark episodes (e.g., washer spin cycles, microwave beeps, fire alarms). The original recordings vary in format (44.1, 48, and 96 kHz; mono and stereo); since SoundSpaces models each emitter as a point source, we downmix stereo files to mono and resample all sources to 24 kHz prior to spatialization.

Spatial rendering. For each synthetic episode, we render room impulse responses (RIRs) with SoundSpaces 2.0 Chen et al. (2022a) on the corresponding HM3D scene Ramakrishnan et al. (2021), setting the rendering sample rate to 24 kHz and using a four-channel first-order ambisonics (FOA) receiver at the agent position. Each mono source is convolved with its FOA RIR at the matched rate, and the resulting episode audio is released as 24 kHz four-channel FOA WAV files.

Real-world recordings. The 272 real-world episodes are captured with a Zoom H3-VR ambisonic microphone at 48 kHz and downsampled to 24 kHz FOA to match the synthetic distribution.

A.2 Details on Benchmark Design

Table A.1: **Benchmark construction by grounding type.** Conditions used to construct easy, medium, and hard instances, along with infeasibility conditions for each grounding type.

Type	Easy	Medium	Hard	Infeasible
G0: Speech	Fluent utterance explicitly naming the device and room.	Speech disfluencies, such as hesitations and repetitions, are added.	Background noise is added to the disfluent utterance.	Named target absent; required actuator unavailable; physical limits; conflicting temporal constraints.
G1: Gesture	One device and one person.	Additional candidate devices are introduced.	Multiple candidate devices and additional non-speaking people.	No device or requested capability matching the pointed target; physical limits; conflicting temporal constraints.
G2: Sound	One candidate device capable of producing sound.	Multiple candidate devices, with only one producing sound.	Sounds from other devices are also present.	Referenced sound absent; physical limits; conflicting temporal constraints.
G3-V: Visual Space	One device and one person.	Additional candidate devices are introduced.	Non-speaking people are added, requiring direction interpretation from the speaker's perspective.	No device with the requested capability in the referenced direction; physical limits; conflicting temporal constraints.

Table continues on the following page

Table A.1 (continued)

Type	Easy	Medium	Hard	Infeasible
G3-A: Auditory Space	One candidate device capable of producing sound and one person.	Additional candidate devices are introduced.	Non-speaking people are added, requiring joint resolution of the sound source, direction, and speaker.	Referenced sound absent; physical limits; conflicting temporal constraints.
G4: Identity	One person referring to "my room".	Non-speaking people are added, requiring speaker identification to resolve "my room".	The speaker points to another person and refers to "his room" or "her room".	Target device or requested capability absent from the person's room; required actuator unavailable; physical limits; conflicting temporal constraints.

We define easy, medium, hard, and infeasible settings within each grounding type by varying candidate devices, distractors, grounding cues, and task feasibility. Table A.1 summarizes these settings.

G0: Speech. Easy uses fluent, explicit references; medium adds disfluencies; hard adds background noise. Infeasible cases involve missing targets or actuators, physical limits, or temporal conflicts.

G1: Gesture. Easy has one device and person; medium adds candidate devices; hard adds non-speakers. Infeasible cases involve missing pointed targets or capabilities, physical limits, or temporal conflicts.

G2: Sound. Easy has one sound-capable device; medium adds candidate devices but only one audible source; hard adds another device emitting a different type of sound, so the agent must disambiguate under a noisy acoustic scene. Infeasible cases involve absent referenced sounds, physical limits, or temporal conflicts.

G3-V: Visual Space. Easy has one device and person; medium adds candidate devices; hard adds non-speakers to test speaker-relative directions. Infeasible cases involve no suitable device in the referenced direction, physical limits, or temporal conflicts.

G3-A: Auditory Space. Easy has one sound-capable device and person; medium adds candidates; hard adds non-speakers to test speaker-relative sound localization. Infeasible cases involve absent referenced sounds, physical limits, or temporal conflicts.

G4: User Identity. Easy uses "my room" with one person; medium adds non-speakers; hard uses pointing with "his/her room". Infeasible cases involve missing devices or capabilities in that person's room, unavailable actuators, physical limits, or temporal conflicts.

A.3 Examples

Synthetic Data Figure A.1 shows one synthetic episode for every valid combination of grounding type (G0–G4) and query type (Q1–Q4). Each episode consists of a single rendered view of the room, paired with a spatial audio recording (binaural and first-order Ambisonics) that contains the user's spoken request and, where relevant, the sound emitted by a device; the transcript printed under each image is *not* given to the agent. The grounding type determines which part of this observation carries the missing information. In G0 the utterance names the device and its room outright, so correct speech recognition suffices. In G1 the utterance is purely deictic ("Play this."), and the target is the device that the speaker's outstretched arm points at. In G2 the device is identified only by the sound it makes ("stop that whirring sound"), so the agent must recognize the sound and localize its source among the devices in view. In G3 the device is described relative to the speaker's own body ("above my head", "on my right"), which requires estimating where the speaker

Table A.2: **Device list.** Device categories and types included in our benchmark.

Category	Devices
Lighting	On/off light, dimmable light
Climate and air quality	Air conditioner, fan, air purifier, dehumidifier
Media	TV, audio player
Kitchen	Refrigerator, freezer, microwave, coffee machine, dishwasher
Laundry and cleaning	Laundry washer, laundry dryer, robot vacuum cleaner
Monitoring	Electrical sensor

stands and which way they face; the direction is either combined with a device sound or stands alone. In G4 the request refers to "my room": the agent has to recognize who is speaking from their face and voice, and then look up which room belongs to that household member, typically a room other than the one being observed. Implicit-intent queries (Q4) state a discomfort rather than a command ("My room is too dark."), and the target is whichever device in the resolved room can relieve it.

Alongside the raw observation, the simulator exposes a *reference image* of the room (Fig. A.2). It is rendered from the same camera without any person, and every controllable device is outlined with a box and an identifier (D1, D2, . . .) that matches the device list returned for that room. The reference image tells the agent *where each device is*, but not *which one the user means*: deciding that the pointing arm, the beeping sound or the speaker's left-hand side coincides with a particular box is exactly the grounding problem the benchmark measures..

Real-World Data Figure A.3 shows one recorded episode per grounding type. Each real-world episode is a short video clip from a fixed wide-angle camera, synchronized with a four-channel recording from a first-order Ambisonics microphone placed at the camera; participants act out the request in furnished living rooms, kitchens and a laundry room that contain real appliances. The episodes follow the same taxonomy as the synthetic set, so the same cues must be read from real footage: the pointing arm in G1 occupies a small part of the wide-angle frame and is held only briefly, the beeping microwave in G2 competes with room reverberation and the speaker's own voice, and "in front of me" in G3 depends on the participant's body orientation rather than on the camera's. The G4 example is an *infeasible* episode: the speaker is recognized as the son, but no device in his room can lower the humidity, so the correct behaviour is to explain that the request cannot be carried out instead of acting on a device in the visible room. As for synthetic episodes, a reference image with device identifiers is provided for every room (bottom row).

A.4 Detail on Home Devices and Matter Cluster

Device list. Our benchmark covers a diverse set of smart-home devices across six categories, as summarized in Tab. A.2. The device set includes common lighting, climate, media, kitchen, cleaning, and monitoring appliances, enabling evaluation across a range of everyday smart-home scenarios.

Matter cluster. We model device states and operations using Matter clusters. Each cluster exposes a set of attributes describing the current device state and, when applicable, commands that can modify the state. Tab. A.3 summarizes the clusters, attributes, and commands supported in our benchmark.

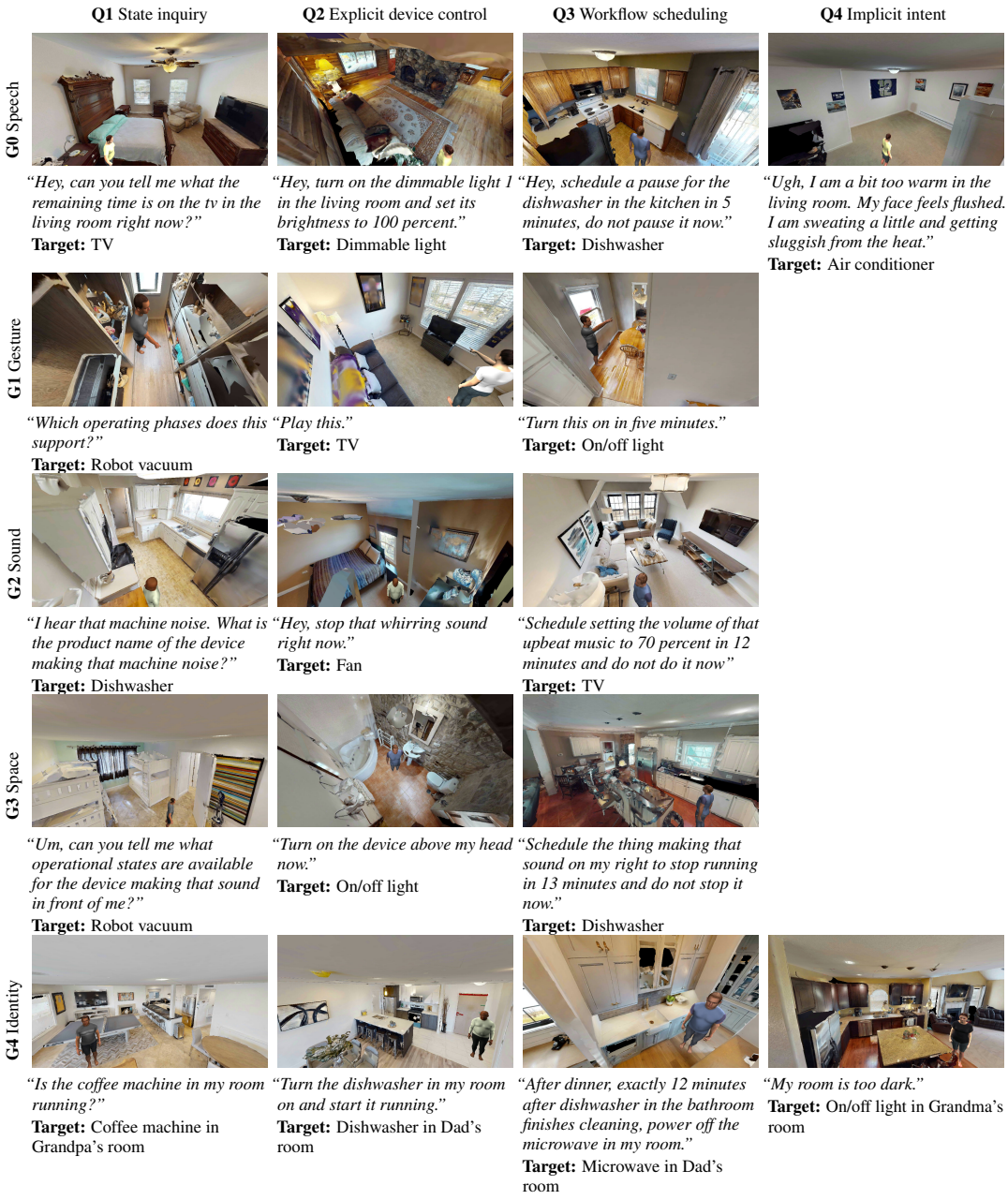


Figure A.1: **Synthetic examples**, one per grounding type (rows) and query type (columns). Each cell shows the rendered observation, the spoken request (delivered to the agent only as spatial audio), and the ground-truth target device. Q4 is instantiated only for G0 and G4.

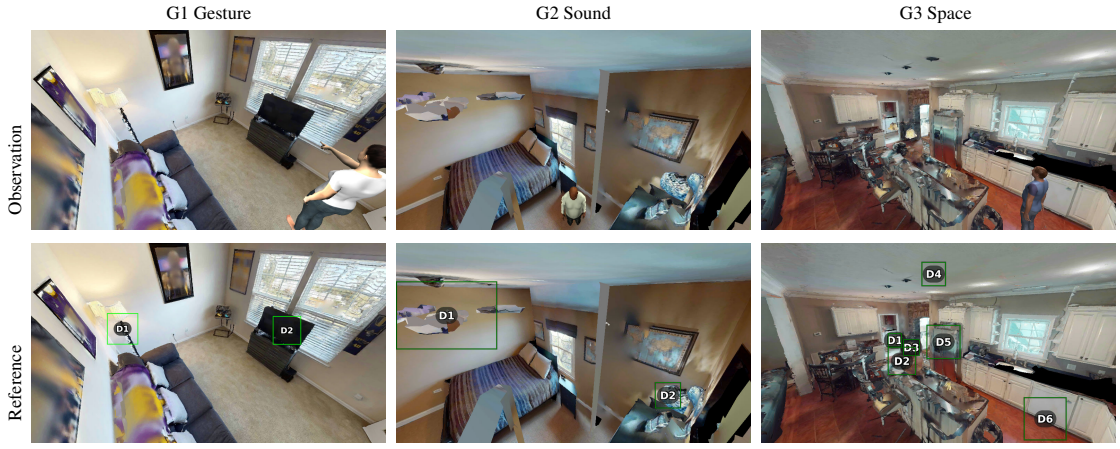
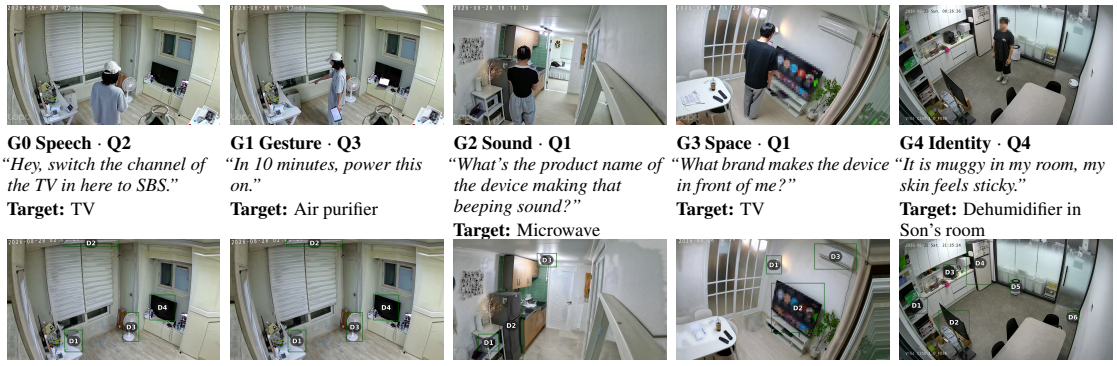


Figure A.2: **Reference images.** Top: the observation the agent receives. Bottom: the reference image of the same room, rendered without people, in which each device carries the identifier used in the device list.



G0 Speech · Q2
"Hey, switch the channel of the TV in here to SBS."
Target: TV

G1 Gesture · Q3
"In 10 minutes, power this on."
Target: Air purifier

G2 Sound · Q1
"What's the product name of the device making that beeping sound?"
Target: Microwave

G3 Space · Q1
"What brand makes the device in front of me?"
Target: TV

G4 Identity · Q4
"It is muggy in my room, my skin feels sticky."
Target: Dehumidifier in Son's room

Figure A.3: **Real-world examples**, one per grounding type. Top: a frame of the recorded clip with the spoken request and the target device. Bottom: the reference image of the same room with device identifiers. The G4 request is infeasible: the target is the device that would have been needed, which the speaker's room does not contain.

Table A.3: **Matter clusters.** Attributes and commands supported by the Matter clusters used in our benchmark.

Cluster	Attributes	Commands
Basic Information	VendorName, VendorID, ProductName, ProductID	–
Descriptor	DeviceTypeList, ServerList, ClientList, PartsList, TagList	–
Identify	IdentifyTime, IdentifyType, FeatureMap, ClusterRevision	Identify, TriggerEffect
On/Off	OnOff, GlobalSceneControl, OnTime, OffWaitTime, StartUpOnOff	On, Off, Toggle
Level Control	CurrentLevel, RemainingTime, MinLevel, MaxLevel, CurrentFrequency, MinFrequency, MaxFrequency, Options, OnOffTransitionTime, OnLevel, OnTransitionTime, OffTransitionTime, DefaultMoveRate, StartUpCurrentLevel	MoveToLevel, Move, Step, Stop, MoveToLevelWithOnOff, MoveWithOnOff, StepWithOnOff, StopWithOnOff, MoveToClosestFrequency
Thermostat	LocalTemperature, OccupiedCoolingSetpoint, OccupiedHeatingSetpoint, SystemMode, ControlSequenceOfOperation	SetpointRaiseLower
Fan Control	FanMode, FanModeSequence, PercentSetting, PercentCurrent	Step
Temperature Measurement	MeasuredValue, MinMeasuredValue, MaxMeasuredValue	–
Temperature Control	TemperatureSetpoint, MinTemperature, MaxTemperature	SetTemperature
Relative Humidity Measurement	MeasuredValue, MinMeasuredValue, MaxMeasuredValue, Tolerance	–
Media Playback	CurrentState	Play, Pause, Stop, StartOver, Previous, Next, Rewind, FastForward
Channel	ChannelList, Lineup, CurrentChannel	ChangeChannel, ChangeChannelByNumber, SkipChannel
Keypad Input	FeatureMap, ClusterRevision	SendKey
Operational State	PhaseList, CurrentPhase, CountdownTime, OperationalStateList, OperationalState, OperationalError	Start, Stop, Pause, Resume, OperationalCommandResponse
Microwave Mode	SupportedModes, CurrentMode	ChangeToMode
Coffee Machine Mode	SupportedModes, CurrentMode	ChangeToMode

(continued)

Table A.3 continues on the following page

Cluster	Attributes	Commands
Dishwasher Mode	SupportedModes, CurrentMode	ChangeToMode
Laundry Washer Mode	SupportedModes, CurrentMode	ChangeToMode
RTCC Mode	SupportedModes, CurrentMode	ChangeToMode
Laundry Washer Controls	SpinSpeeds, SpinSpeedCurrent, NumberOfRinses, SupportedRinses	–
Laundry Dryer Controls	SupportedDrynessLevels, SelectedDrynessLevel	–
RVC Run Mode	SupportedModes, CurrentMode, FeatureMap, ClusterRevision	ChangeToMode
RVC Clean Mode	SupportedModes, CurrentMode, FeatureMap, ClusterRevision	ChangeToMode
RVC Operational State	PhaseList, CurrentPhase, CountdownTime, OperationalStateList, OperationalState, OperationalError, ClusterRevision	Pause, Resume, GoHome
Alarm	AlarmActive, AlarmCode	Trigger, Silence
Dishwasher Alarm	Mask, Latch, State, Supported	ModifyEnabledAlarms, Reset
Electrical Power Measurement	PowerMode, NumberOfMeasurementTypes, Accuracy, ActivePower, FeatureMap, ClusterRevision	–
Electrical Energy Measurement	Accuracy, FeatureMap, ClusterRevision	–
Power Topology	FeatureMap, ClusterRevision	–

B Details on Tools

B.1 Original Tools

Tab. B.1 summarizes the smart-home tools available to the agent for querying environment states (Seo et al., 2026), inspecting Matter device structures, executing device operations, and managing scheduled workflows. We additionally introduce `get_users` and `declare_target_device` for OmniSmartHome. The former provides household user information and enrolled identity references required for identity grounding, while the latter records the device selected by the agent as the grounded target.

B.2 Perception Tools

Tab. B.2 lists the perception tools of PROME used to gather information distributed across different modalities in the current episode.

Table B.1: **Smart-home tools.** † denotes tools additionally introduced for OmniSmartHome.

Tool	Args.	Output
get_rooms	–	Room IDs and names
get_room_devices	(room_id)	Devices in the specified room
get_room_states	(room_id)	Environmental states of the specified room
get_device_structure	(device_id)	Device endpoints, clusters, attributes, and commands
get_all_attributes	(device_id)	All attribute values of the specified device
get_attribute	(device_id, endpoint_id, cluster_id, attribute_id)	Requested attribute value
get_cluster_doc	(query, top_k)	Relevant Matter cluster documentation
get_environment_control_rules	(state)	Rules governing the specified environmental state
execute_command	(device_id, endpoint_id, cluster_id, command_id, command_args)	Command execution result
write_attribute	(device_id, endpoint_id, cluster_id, attribute_id, attribute_value)	Attribute update result
get_current_time	–	Current simulation time
schedule_workflow	(start_time, steps)	Created workflow ID and scheduling result
get_workflow_status	(workflow_id)	Workflow status
get_workflow_list	–	Registered workflows
cancel_workflow	(workflow_id)	Cancellation result
finish	(answer)	Final answer and episode termination
get_users†	–	Household user profiles and enrolled identity references
declare_target_device†	(device_id)	Recorded target-device declaration; "none" indicates no matching device

For audio, we use Whisper large-v3 (Radford et al., 2023) for speech transcription, OpenVocabularySELD (Shimada et al., 2025) for sound-event localization, and WavLM (Chen et al., 2022b) for speaker verification against enrolled household voices.

For visual perception, we use Keypoint R-CNN (He et al., 2017) for person detection, RTMW (Jiang et al., 2024) for pointing estimation, DirectMHP (Zhou et al., 2023) for head-pose estimation, and DINOv2 (Oquab et al., 2024) for face matching against enrolled household face images. Together, these tools help the agent gather multimodal information.

C Additional Results on OmniSmartHome

C.1 Baseline Models

We evaluate a diverse set of open-source and proprietary Omni-LLMs spanning different model scales and architectures. **Small open-source models:** Video-LLaMA2 (Cheng et al., 2024), Qwen2.5-Omni (Xu et al., 2025a), Spatial-Omni (Zhu et al., 2026), video-SALMONN 2+ (Tang et al., 2025), video-SALMONN-o1 (Sun et al., 2025), OmniVinci (Ye et al., 2026), and MiniCPM-o-4.5 (Cui et al., 2026). **Large open-**

Table B.2: **Perception tools used by the agent.** Each tool processes the current episode observation, optionally together with enrolled household information, and returns structured perceptual evidence.

Tool	Model	Args.	Perceptual input	Output
asr	Whisper large-v3	-	Episode audio	Speech transcript.
estimate_doa	OpenVocabularySELD	-	Episode audio	Sound-event labels and coarse horizontal directions: left, middle, or right.
people_detection	Keypoint R-CNN	-	Episode image/video	Person labels and bounding boxes.
estimate_pointing	RTMW	people_label	Episode image/video	Hand bounding box, visibility, and an annotated image.
estimate_head_pos	DirectMHP	people_label	Episode image/video	Facing directions: front_toward, their_right_toward, and their_left_toward.
match_face	DINOv2	people_label	Episode image/video, enrolled household face images	Selected person label and similarity scores indexed by user_id.
verify_speaker	WavLM	candidate_user_id	Episode audio, enrolled household voices	Candidate user ID and speaker similarity score.

source models: Gemma 4 (Team et al., 2026), Nemotron-3-Nano-Omni (Deshmukh et al., 2026), Qwen3-Omni-Instruct/Thinking (Xu et al., 2025b), and MiMo-V2.5 (mim, 2026). **Proprietary models:** Gemini 2.5 Flash/Pro (Comanici et al., 2025), Gemini 3.1 Pro (Preview) (Gemini Team, Google DeepMind, 2026), and Qwen3.8-Omni-Flash (Qwen Team, 2026). Among these, only Spatial-Omni and Qwen3.8-Omni-Flash natively accept spatial (FOA) audio; for all other models, we downmix the FOA recordings to mono before inference. Proprietary models are accessed through their official APIs (Vertex AI for Gemini, Alibaba Cloud for Qwen3.8-Omni-Flash). Among the open-source models, MiMo-V2.5 is accessed through OpenRouter due to their scale, and the rest are run locally with their released checkpoints.

C.2 Additional Results by Query Type

Table C.1 reports accuracy by query type. For large open-source Omni-LLMs and proprietary models, grounding accuracy remains relatively stable across query types, with the exception of intent queries (Q4). We attribute this drop to the additional reasoning Q4 demands, namely interpreting the user's intent and selecting an appropriate device, rather than to multimodal perception itself; for the remaining types, resolving the multimodal reference is comparably difficult regardless of the query type. Goal accuracy, in contrast, varies substantially across query types. Consistent with the findings of Seo et al. (2026), models achieve particularly low goal accuracy on queries that require scheduling (Q3) and implicit intent inference (Q4), indicating that these capabilities remain challenging for current LLM-based agents. PROME also yields consistent gains in both grounding and goal accuracy across all query types.

Grounding is a prerequisite for goal completion. Table C.2 decomposes goal accuracy by whether the agent declared the correct target device. Two patterns emerge. First, when grounding fails, goal accuracy collapses: $P(L \mid \neg G)$ stays below 12% for every model and below 5% for most, so a wrong referent is almost never recovered downstream. Second, correct grounding is necessary but not sufficient, and how much of it converts into goal success depends strongly on model capacity. Small open-source models complete the goal in only 32–49% of the episodes they ground correctly, as they often fail at the subsequent multi-step device operation and scheduling even with the right target in hand. Larger open-source models reach 65–73%, and proprietary models 75–84%. Hence, for weaker models both grounding and post-grounding execution are bottlenecks, whereas for stronger models the remaining errors are increasingly dominated by grounding itself.

Table C.1: **Results on OmniSmartHome across query types (Q1–Q4).** For each query type, we report goal accuracy (Goal) and grounding accuracy (Gr.), averaged over grounding types. For our agentic system, the (Gain) row reports absolute changes in percentage points relative to the corresponding vanilla backbone.

	Q1 State Inquiry		**Q2** Control		**Q3** Schedule		**Q4** Intent	
Model	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal
Open-source Omni-LLMs (<10B)								
Video-LLaMA2 7B	6.0	9.2	5.2	0.6	5.8	6.7	6.8	0.5
Qwen2.5-Omni 7B	31.2	32.9	36.2	8.8	30.4	13.3	35.9	12.0
Spatial-Omni 7B	18.3	20.4	23.1	2.5	19.8	15.0	16.7	2.6
video-SALMONN2+ 7B	32.3	24.6	24.8	6.0	23.1	14.0	29.2	5.2
video-SALMONN-o1 7B	24.4	16.0	27.1	5.6	23.8	12.9	20.3	1.0
OmniVinci 9B	16.9	17.5	17.7	1.2	16.7	10.2	22.9	2.1
MiniCPM-o-4.5 9B	61.3	41.7	63.7	32.1	60.6	20.6	51.0	27.1
Large Open-source Omni-LLMs								
Gemma4 12B	50.4	22.3	49.4	24.6	56.0	27.5	40.1	28.1
Nemotron3-Omni 30B-A3B	63.7	62.1	67.7	41.9	60.0	36.2	54.2	28.1
Qwen3-Omni-Instruct 30B-A3B	72.1	66.2	71.9	42.7	59.6	26.9	57.3	32.3
Qwen3-Omni-Think 30B-A3B	68.8	65.4	72.7	49.6	59.0	34.6	58.9	37.5
MiMo-V2.5 310B-A15B	74.6	73.3	81.7	64.6	82.7	46.9	69.3	50.0
Proprietary Omni-LLMs								
Gemini-2.5-Flash	72.5	69.0	77.9	59.2	77.9	46.5	55.2	41.7
Gemini-2.5-Pro	80.4	82.1	83.5	59.8	84.6	50.0	67.2	56.8
Gemini-3.1-Pro (Preview)	83.5	84.2	89.4	76.2	91.9	64.8	65.6	60.9
Qwen3.8-Omni-Flash	81.2	81.0	85.4	63.1	86.0	50.6	63.0	49.5
Agent Baseline								
Qwen2.5-Omni + **PROME**	39.4	31.9	41.2	15.8	34.6	21.0	37.0	18.8
	+8.1	-1.0	**+5.0**	**+7.1**	**+4.2**	**+7.7**	**+1.0**	**+6.8**
Gemma4 + **PROME**	72.5	66.9	80.0	66.0	75.4	48.8	59.4	49.5
	+22.1	**+44.6**	**+30.6**	**+41.5**	**+19.4**	**+21.2**	**+19.3**	**+21.4**
Nemotron3-Omni + **PROME**	66.7	63.7	70.8	52.9	61.7	37.9	56.2	31.2
	+2.9	**+1.7**	**+3.1**	**+11.0**	**+1.7**	**+1.7**	**+2.1**	**+3.1**
Qwen3-Omni-Think + **PROME**	72.9	69.6	76.9	58.5	65.6	40.8	57.8	35.4
	+4.2	**+4.2**	**+4.2**	**+9.0**	**+6.7**	**+6.2**	-1.0	-2.1
Gemini-2.5-Flash + **PROME**	80.6	77.1	81.5	66.5	80.0	49.4	59.9	46.9
	+8.1	**+8.1**	**+3.5**	**+7.3**	**+2.1**	**+2.9**	**+4.7**	**+5.2**
Gemini-2.5-Pro + **PROME**	85.0	85.0	87.1	70.8	86.5	52.9	76.6	66.1
	+4.6	**+2.9**	**+3.5**	**+11.0**	**+1.9**	**+2.9**	**+9.4**	**+9.4**

Table C.2: **Goal accuracy conditioned on the grounding outcome.** $P(L \mid G)$ is the goal accuracy when the declared target device is correct, and $P(L \mid \neg G)$ when it is incorrect.

Model	$P(L \mid G)$	$P(L \mid \neg G)$
Open-source Omni-LLMs (<10B)		
Video-LLaMA2 7B	43.5	2.7
Qwen2.5-Omni 7B	33.8	9.6
video-SALMONN2+ 7B	35.5	5.8
video-SALMONN-o1 7B	24.8	3.7
OmniVinci 9B	32.4	3.7
MiniCPM-o-4.5 9B	49.1	3.0
Open-source Omni-LLMs		
Gemma4 12B	49.0	0.9
Nemotron3-Omni 30B-A3B	68.5	4.3
Qwen3-Omni-Instruct 30B-A3B	64.9	1.7
Qwen3-Omni-Think 30B-A3B	72.8	1.3
MiMo-V2.5	74.3	4.0
Proprietary Omni-LLMs		
Gemini-2.5-Flash	74.6	4.9
Gemini-2.5-Pro	75.9	8.4
Gemini-3.1-Pro (Preview)	**83.8**	11.9
Qwen3.8-Omni-Flash	75.5	7.7

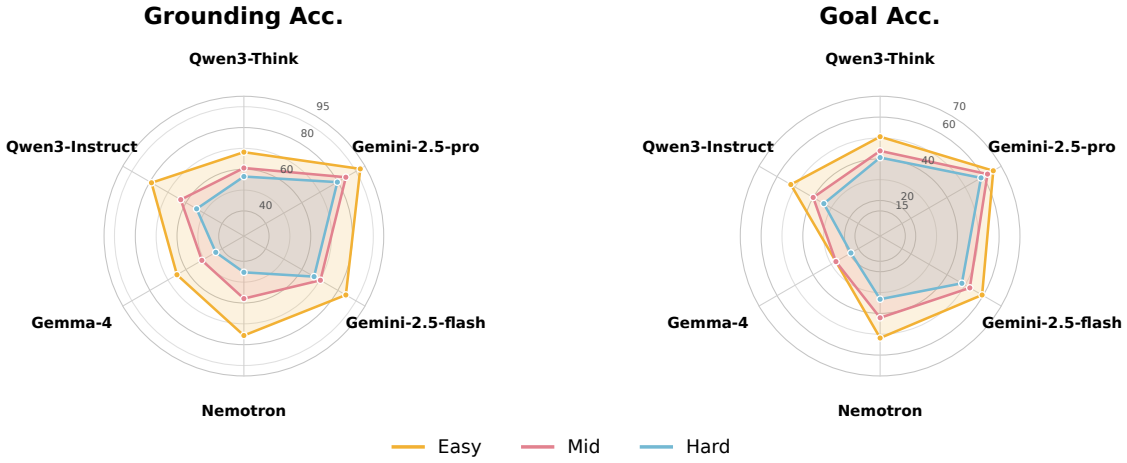


Figure C.1: **Model performance across grounding difficulty levels for feasible queries.** The *easy*, *mid*, and *hard* levels are defined according to the type and number of distractors. Both grounding accuracy and goal accuracy consistently decrease as grounding difficulty increases.

C.3 Additional Results by Difficulty Level

Our synthetic dataset, FEASIBLE, is categorized into three levels of *grounding difficulty*: *easy*, *mid*, and *hard*. These difficulty levels are defined based on the composition of distractors in each instance, including their

types and quantities. As the distractors become more numerous or more difficult to distinguish from the target evidence, the grounding task becomes correspondingly more challenging.

Figure C.1 presents model performance across the three difficulty levels. We observe a clear and consistent ordering for both *grounding accuracy* and *goal accuracy*: *easy* > *mid* > *hard*. This pattern holds broadly across models, showing that increasing grounding difficulty not only makes it harder to identify the correct evidence, but also leads to lower downstream task success. The results further validate that our difficulty stratification captures meaningful differences in the level of grounding challenge induced by distractors.

C.4 Modality Ablation

To assess whether multimodal observations genuinely contribute to grounding in OmniSmartHome, we evaluate a text-only variant of the vanilla agent, removing all visual and acoustic observations and providing the user request only as a transcript. We distinguish the *type* of a device, i.e., its category (e.g., dimmable light), from its *instance*, i.e., a particular device of that type (e.g., dimmable_light_1 or dimmable_light_2). Fig. C.2 reports the grounding accuracy of Gemini-2.5-Flash, separating cases where the target is the only instance of its type in the target room from those where multiple devices share the same type.

The results reveal a clear division of labor between language and multimodal evidence: language is often sufficient to identify the *type* of the referent, whereas resolving its *instance* requires audio-visual context. For G0, the spoken request itself contains all information necessary for grounding, and replacing it with its transcript therefore causes almost no degradation. For G1–G4, however, multimodal evidence becomes increasingly important, especially when multiple devices share the target type. When the target is the only instance of its type, linguistic cues can still narrow the candidate set. In contrast, when two or more same-type candidates are present, language alone cannot distinguish among them, forcing grounding to rely on audio-visual evidence and resulting in sharp drops (G1: 27.8→2.8, G2: 100.0→81.2, G3: 59.1→0.0). G4 degrades substantially in both settings because grounding requires the agent to first identify the speaker, then infer the corresponding room, and finally locate the device; without multimodal observations, this reasoning chain already breaks at its first step, irrespective of the number of candidate devices.

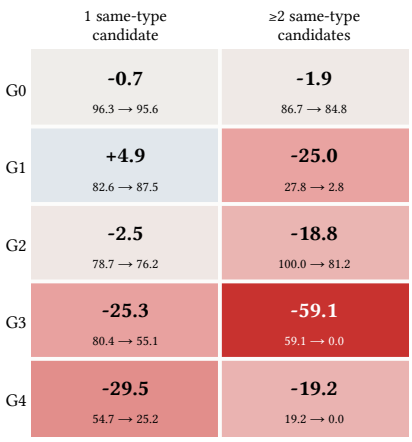


Figure C.2: **Modality ablation.** Grounding accuracy of Gemini-2.5-Flash with the full multimodal input and text-only input.

C.5 Error Analysis

Fig. 5 reports one representative failure mechanism per grounding type for Qwen3-Omni-Think, Gemma4-12B, Gemini-2.5-Flash, and Gemini-2.5-Pro, in the base agent setting. All four panels are computed from recorded episode traces (tool calls, observations, and the declared device) together with the benchmark ground truth; no additional model calls are made. An episode counts as a *grounding failure* if the declared device is not the referent or no device is declared, and as a *wrong pick* if a device other than the referent is declared. Episodes that ended in a protocol or infrastructure failure (malformed output, timeout, context overflow) are excluded throughout.

G1. Among feasible synthetic episodes, we examine wrong picks. Candidates are the in-frame device bounding boxes returned by `get_room_devices`. The bar shows the share of wrong picks in which the declared device is the candidate closest, in image space, to the user's hand or body (which coincide in most rooms).

G2. Among feasible synthetic and real-world episodes, we again examine wrong picks. The bar shows the share of wrong picks in which the agent declared the device *without reading the state of any device* beforehand, i.e., no state-reading tool (`get_attribute`, `get_device_structure`, `get_all_attributes`) was called on any device before the declaration. Such declarations rely on the room layout and the sound alone and never verify that the chosen device is in the described state.

G3. We consider feasible synthetic episodes whose referent is described as "on my left" or "on my right." The user's orientation is measured as the angle between their body-forward vector and the vector from the user to the camera, taken from the scene layout. Episodes with an angle below $80°$ are *facing camera* (n=12), where the user's left appears on the image's right and left/right must be mirrored; those above $100°$ are *facing away* (n=18), where image and user sides coincide. Episodes in between are excluded. The bars show grounding failure rates within each group.

G4. Among feasible synthetic and real-world episodes, we examine grounding failures. The bar shows the share of failures in which the agent never called a speaker- or face-identification tool (`get_users`, `match_face`, `verify_speaker`, `people_detection`) before declaring, i.e., it committed to a device without attempting to identify the speaker.

D PROME

D.1 Details of PROME

PROME adds a small procedural memory $\mathcal{M} = \{\pi_\sigma\}_{\sigma\in\Sigma}$, a bank of grounding procedures indexed by the grounding schema registry Σ.

Grounding schema. The grounding schema registry Σ is mined from the synthetic training episodes using only the audio-visual observations of each episode; target labels, grounding-type labels, and the home state are never shown. Mining proceeds in two passes with an Omni-LLM (Qwen3-Omni-Think (Xu et al., 2025b)). In the first pass, the model describes the reusable cues of each episode in one sentence, and is forbidden to name any device, room, person, tool, or tool sequence. In the second pass, episodes are visited in random order while the table is grown incrementally: for each episode, the model either assigns it to an existing grounding schema or adds a new one, judging sameness by the kind of evidence relation required to identify the target rather than by wording or by the downstream request. This yields $|\Sigma| = 8$ grounding schemas (Tab. D.1). Since they are induced automatically, they do not correspond one-to-one to our predefined grounding types.

At test time, Σ is frozen. Given the request and the initial observation, the agent retrieves $\sigma^* \in \Sigma$ once per episode by choosing among the eight entries.

State. Progress within an episode is summarized by a discrete agent state $s_t = (p_t, c_t, r_t) \in \mathcal{S}$ with $|\mathcal{S}| = 5 \times 3 \times 4 = 60$:

- $p_t \in \{\texttt{prelist}, \texttt{unique}, \texttt{multi}, \texttt{absent}, \texttt{unknown}\}$: candidate plausibility—whether the room has been listed yet and, if so, whether one, several, or no listed devices match the requested type;
- $c_t \in \{0, 1, 2+\}$: how many candidates have had their state read through tools such as `get_attribute`;
- $r_t \in \{\texttt{none}, \texttt{spatial}, \texttt{identity}, \texttt{both}\}$: which class of perception tool has been called.

The state is updated deterministically after each ReAct step: LIST tools such as `get_room_devices` or `get_users` set p_t, READ tools increment c_t, and perception tools set r_t (action classes are defined below).

Table D.1: **The grounding schema registry.** Eight grounding schemas were induced from the synthetic training episodes by the table-construction prompt (Appendix E.1) and frozen; at inference, the selection prompt picks exactly one per episode. Each σ names the evidence relation that identifies the target device.

ID	Evidence relation	Grounding Schema
σ_1	pointing gesture	Identifying a specific device in a room image by a pointing gesture when the user refers to it as 'this' or 'that' to request information about it.
σ_2	position rel. to user	Identifying a specific device in a room image by its position relative to the user's body (e.g., above head, to the left) when the user refers to it with such spatial descriptors.
σ_3	audible sound	Identifying a specific device in a room image by its audible sound when the user refers to it as the source of that sound to request information about it.
σ_4	device + room named	Identifying a specific device in a room image by the user explicitly naming the device and room to request an action on it.
σ_5	household member	Identifying a specific household member from a room image to resolve pronoun references (e.g., 'his' or 'her') for room-based requests.
σ_6	room named	Identifying a specific device in a room image by the user explicitly naming the room to request an action on a device within that room.
σ_7	sound + position	Identifying a specific device in a room image by both the audible sound it is producing and its position relative to the user's body to request information about it.
σ_8	room state	Identifying a specific device in a room image by the user describing a room state (e.g., air quality, temperature) to determine which device can modify that state.

Action and guidance. The action set is $\mathcal{A} = \{\text{LIST}, \text{READ}, \text{PERC:}\langle\text{tool}\rangle, \text{DECLARE}\}$, and every tool the agent may call before declaring maps to exactly one entry:

- LIST: room, home, and household-roster queries (e.g., `get_room_devices`, `get_users`);
- READ: attribute and structure queries on a device (e.g., `get_attribute`);
- PERC:$\langle$tool$\rangle$: one of the seven perception tools in Tab. B.2 (e.g., `people_detection`, `estimate_doa`);
- DECLARE: declaring the target device.

Tools used after device declaration, such as command execution and scheduling, are outside $\mathcal{A}$, and no guidance is injected once the target has been declared. The selected action a_t is verbalized as a [PROCEDURAL GUIDANCE] inserted before the agent's next reasoning step. The guidance first states the current state ("Now: the devices are listed but the candidates are not narrowed yet; no candidate state read yet; no perception tool used yet.") and then suggests the evidence to gather next ("read the candidates' state" or "call `estimate_doa` to acquire the missing evidence"). The guidance names only the type of evidence to gather, never a candidate device.

Initial policy. The initial logits θ^0 combine a cost prior shared by all grounding schemas with a utility bonus Δ for perception tools:

$$\theta^0_{\sigma,s,a} = -\gamma\,\text{cost}(a,s) + \Delta_{\sigma,s,a}, \qquad \gamma = 3, \tag{2}$$

where $\Delta_{\sigma,s,a} = 0$ for every non-perception action.

Action	Cost
LIST	0 before the room is listed; 10 after
READ	0 while unresolved; 0.5 after one verifying read of a unique candidate
PERC:⟨tool⟩	1
DECLARE	0 once a unique candidate is verified; 0.5 if unverified, or if 2+ candidates were read without a match; 1 otherwise

The cost prior encodes a general-purpose procedure: list the room, check the current states of candidate devices, and declare, calling a perception tool only when the ambiguity cannot be resolved from the Omni-LLM's own perception of the initial observation. Since each perception tool queries an external expert model and thus adds latency, the cost prior generally discourages such calls.

The utility bonus Δ estimates, per (σ, s) and perception tool a, whether invoking the tool there actually helps. We roll out the memory-free agent on the training episodes twice: a *vanilla* rollout τ^-_{ref} without perception tools and a *perception-only* rollout τ^+_{ref} with them, i.e., the reference rollouts of Eq. (1). Whenever the perception-only rollout of episode e invokes tool a at (σ_e, s), we record the same-episode gain $d = G(\tau^+_{\text{ref}}) - \hat{r}^-(e)$, where $G \in \{0, 1\}$ is the grounding reward of Eq. (1) and $\hat{r}^-(e)$ is the vanilla success rate on that episode. With $\bar{d}$ the mean gain over the collected pairs,

$$\Delta_{\sigma,s,a} = \begin{cases} \text{clip}(\lambda \bar{d}, -\kappa, \kappa) & \text{if } \#\text{pairs} \geq n_{\min} \text{ and } |\bar{d}|/\text{SE}(d) \geq z_{\min}, \\ 0 & \text{if } \#\text{pairs} \geq n_{\min} \text{ but not significant}, \\ -\kappa & \text{if } \#\text{pairs} < n_{\min}, \end{cases}$$

with $\lambda = 15$, $\kappa = 3$, $n_{\min} = 10$, $z_{\min} = 1$. Tools that are never used in training are excluded from the support of π_σ.

Policy update. We refine the initial logits θ^0 into θ with the single batched update of Eq. (1). For each training episode e, the group $\mathcal{G}_e = \{\tau, \tau^+_{\text{ref}}, \tau^-_{\text{ref}}\}$ consists of a new memory-guided rollout τ, whose actions are sampled from the initial policy $\pi^0_{\sigma_e}$ induced by θ^0, together with the two reference rollouts already collected for Δ. In practice, $G(\tau)$ combines a binary grounding term with small shaping terms:

$$G(\tau) = \mathbb{1}[\text{correct}] + 0.1 \cdot \mathbb{1}[\text{valid declaration}] + w_{\text{eff}}\, \mathbb{1}[\text{correct}]\, \frac{\min_{\tau' \in \mathcal{G}_e} |\tau'|}{|\tau|} - 0.04 \cdot \#\text{perception calls},$$

with $w_{\text{eff}} = 0.01$ and $|\tau|$ the number of steps. The binary term dominates; the remaining terms favor valid, short, and perception-frugal trajectories. The advantage $A(\tau) = R(\tau) - b_e$ uses the group mean $b_e = \frac{1}{|\mathcal{G}_e|} \sum_{\tau' \in \mathcal{G}_e} R(\tau')$ as the baseline, as in Eq. (1).

In practice, Eq. (1) is instantiated with three additions that stabilize a single update from one batch. (i) *Per-entry normalization*: the gradient of each logit is normalized by its visit count, so that $\theta_{\sigma,s,a}$ is shifted by the mean advantage $\bar{A}_{\sigma,s,a}$ of the rollouts that executed a at (σ, s); for a tabular softmax policy this corresponds to the natural-gradient form of Eq. (1). (ii) *Clipping*: the shift is clipped to $[-c, c]$ to bound the deviation from π^0_σ. (iii) *Significance gating*: as for Δ, the shift is applied only to entries executed by at least $n'_{\min} = 8$ rollouts and whose mean advantage is statistically distinguishable from 0.

Inference. The resulting $\mathcal{M}$ is frozen for evaluation. After each ReAct step, the agent state is updated to s_t and the next action is chosen from $\pi_{\sigma^*}[s_t]$. During training rollouts we sample $a_t \sim \pi^0_{\sigma_e}[s_t]$ to explore alternative evidence-gathering orders; at evaluation we decode greedily, $a_t = \arg\max_a \pi_{\sigma^*}[s_t](a)$.

D.2 REAL-WORLD TRANSFER

In Sec. 4.2, we reported only grounding accuracy for the real-world transfer experiment of PROME. We additionally report goal accuracy in Fig. D.1. Consistent with the grounding results, the memory constructed

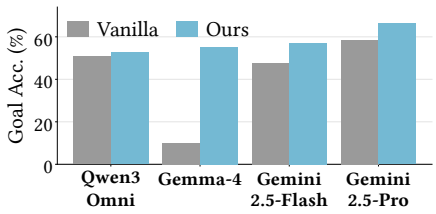


Figure D.1: **Real-world transfer performance.** In addition to grounding accuracy, we report goal accuracy for PROME in the real-world setting. Despite using memory constructed solely from synthetic training data, PROME improves both grounding and goal accuracy in real-world environments.

Table D.2: **Effect of perception quality** on Qwen3-Omni-Think (synthetic split). *Predicted*: off-the-shelf perception tools; *Oracle*: tools return ground-truth attributes from the simulator. Best in **bold**, second best underlined.

Method	Gr.	Goal
Qwen3-Omni-Think	66.4	47.9
+Predicted tools	65.8	47.1
+Oracle tools	69.5	51.1
+PROME	**70.9**	**54.1**

solely from synthetic training data also improves goal accuracy in real-world environments, indicating that the acquired knowledge transfers effectively beyond the synthetic domain.

D.3 Oracle Tool Calling

Tool calling in the main paper relies on off-the-shelf visual and audio perception models, so part of the remaining gap could be attributed to perception errors rather than to the agent itself. To isolate this factor, we define *oracle* perception tools that return ground-truth attributes directly from the simulator—person locations, head pose, and sound source positions—which is possible only because the benchmark is synthetically generated.

This experiment is therefore restricted to the synthetic split; the oracle tools cannot be instantiated on real-world data. Tab. D.2 reports the results on Qwen3-Omni-Think. Replacing predicted perception with oracle perception improves grounding from 65.8 to 69.5 and goal accuracy from 47.1 to 51.1, confirming that perception noise accounts for part of the gap. However, even with perfect perception, oracle tool calling remains below PROME (70.9 / 54.1), which uses only the predicted tools. The gain of PROME thus does not come from better perception alone but from how perceptual evidence is organized and used; the goal-accuracy gap (+3.0) in particular indicates that oracle observations by themselves are insufficient for correct action selection.

D.4 Comparison with other methods

Tab. D.3 compares our approach with two representative agent memory methods, Agent Workflow Memory (AWM) (Wang et al., 2025) and MemP (Fang et al., 2026). AWM induces reusable workflows from past agent trajectories and selectively provides them to guide subsequent actions, while MemP further develops procedural memory through explicit memory construction, retrieval, and update mechanisms. Across both Qwen2.5-Omni and Qwen3-Omni-Think, our method achieves the highest grounding and goal accuracy. In particular, our method shows clear improvements on G2 (Sound), G3 (Space), and G4 (Identity), where resolving the request requires grounding ambiguous references in multimodal evidence. These results indicate

Table D.3: **Comparison with agent baselines on OmniSmartHome.** We report grounding accuracy (Gr.) and goal accuracy (Goal) for each grounding type.

	All		**G0** Speech		**G1** Gesture		**G2** Sound		**G3** Space		**G4** Identity	
Method	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal	Gr.	Goal
Qwen2.5-Omni												
Vanilla	33.0	17.6	55.5	25.5	30.6	19.4	28.1	**14.2**	27.1	17.4	20.3	10.9
AWM	36.0	16.9	**64.8**	31.5	38.9	16.3	26.4	12.2	25.3	11.5	20.3	10.2
MemP	35.9	14.3	63.8	20.8	28.8	15.6	30.9	12.2	27.4	14.6	**23.4**	8.1
PROME	**38.2**	**22.4**	61.5	**34.6**	**41.0**	**28.1**	**35.1**	**14.2**	**28.1**	**22.9**	22.9	**11.7**
Qwen3-Omni-Think												
Vanilla	65.9	48.4	87.8	67.4	66.0	56.6	79.9	56.9	72.2	55.2	28.6	11.7
AWM	64.1	46.8	**89.3**	66.9	64.6	53.1	70.1	50.3	72.6	56.2	27.6	12.0
MemP	64.5	47.5	86.5	66.7	**70.5**	57.6	72.9	51.4	70.8	54.9	27.1	12.5
PROME	**70.2**	**53.9**	87.5	**69.8**	70.1	**58.3**	**86.5**	**68.8**	**77.1**	**62.2**	**35.4**	**17.2**

Table D.4: **Interaction overhead of PROME.** Average number of ReAct steps for vanilla and PROME across models.

Model	Vanilla	PROME	Δ
Gemma-4 12B	6.39	8.54	+2.15
Nemotron3-Omni 30B-A3B	4.90	5.51	+0.62
Qwen3-Omni-Think 30B-A3B	4.57	5.18	+0.61
Gemini-2.5-Flash	5.22	7.01	+1.78
Gemini-2.5-Pro	6.06	7.95	+1.90
Average	5.43	6.84	+1.41

that our procedural memory is better suited to OmniSmartHome than directly applying agent memory methods that are not designed for smart-home settings.

D.5 Latency Analysis

We analyze the additional interaction overhead introduced by PROME by measuring the number of ReAct steps required relative to the vanilla baseline. As shown in the Tab. D.4, PROME incurs only a modest increase in the number of steps, requiring on average 1.41 additional ReAct steps across models. This suggests that the performance gains of PROME come with limited additional inference overhead.

E LLM PROMPT

E.1 GROUNDING SCHEMA REGISTRY CONSTRUCTION PROMPT

Prompt

System Prompt

You maintain a compact registry of reusable smart-home grounding situations.

Inspect the current request, initial image, and initial audio, then compare its grounding situation with the existing lessons as **semantic categories**, not by exact wording. If it overlaps an existing grounding situation, select that lesson. Create a new lesson **only** when the current request requires a completely different kind of grounding ambiguity.

Judge sameness by the **kind of evidence relation** needed to identify the target, rather than surface details or the downstream request. Different values of the same relation generally belong together; a genuinely different combination of evidence relations may be a new situation. Use the task description, image, and audio together.

A new lesson must describe **target identification only**, in one clear general sentence, without concrete names, tool names, or a tool sequence. Return only the requested JSON.

User Prompt

- **Existing lessons:** `{lesson_id: lesson}` for every lesson currently in the registry (empty at the start).
- **Request:** `{query}`
- **Initial observation:** the room image and the recording of the request, attached as image and audio inputs.

Output (JSON only): either `{"lesson_id": <existing id>}` or `{"new_lesson": "<one general sentence>"}`.

E.2 Q1 FEASIBLE JUDGE PROMPT

Prompt

System Prompt

You are a strict evaluator for smart-home LLM agents answering a question about **ONE device** that the user identified only by the sound it makes or by where it sits relative to them—never by the device name, type, or room.

Return **EXACTLY ONE character**: `A` (pass) or `B` (fail). No other text.

WHAT TO JUDGE

- The user asked for the value of a single device attribute.
- Judge **ONLY** whether the agent's Final Answer reports the correct value for that attribute.

CORRECTNESS

- The reported value must match the Expected Value provided below. Equivalent phrasings, formatting, and Matter unit conversions are acceptable (e.g., `2300 == 23.0 °C`, an enum name == its integer code, `True == "on"`, `"LG Electronics" == "LG"`).
- The answer must be **grounded**: the agent must have actually retrieved the value with some tool, not guessed it.

DO NOT PENALIZE

- Which tool the agent used to locate the device or read the value (`get_room_devices`, `find_devices_by_cluster`, `get_device_structure`, `get_attribute`—all acceptable).
- Whether the agent mentions the room name or the device name (the user never gave a room or a name).
- Extra explanation or situational context in the answer, as long as the correct value is clearly stated.

Output `A` only if the value is correct **AND** grounded in an observation; otherwise output `B`.

User Prompt

- **User Query:** `{query}`
- **Target attribute(s) and the expected correct value(s):** `{expected_values}`
- **Agent's ReAct Steps (tool calls and observations):** `{react_steps}`
- **Agent's Final Answer:** `{final_answer}`

EVALUATION STEPS

1. Read the expected value(s) for the asked attribute(s).
2. Check the agent's tool observations actually contain that value for the target device (grounding).
3. Check the Final Answer states that value correctly (allow equivalent phrasings and unit conversions).
4. Output `B` only if the value is wrong, missing, or fabricated. Ignore tool choice, room-name mention, and device-name mention.

Your Decision: `A or B`

E.3 INFEASIBLE JUDGE PROMPT

Prompt

System Prompt

You are a strict evaluator for smart-home LLM agents. The person's request in this episode was **INFEASIBLE** for one specific, labelled reason, and the agent was expected to recognise this, say that it cannot (fully) carry out the request, and give that reason.

Return **EXACTLY ONE character**: `A` (pass) or `B` (fail). No other text.

Pass (`A`) ONLY IF the agent's Final Answer meets ALL conditions:

1. **Refusal:** it clearly states that the request cannot be carried out / was not carried out (a partial or hedged "I did it but . . . " is not a refusal).
2. **Right Reason:** the reason it gives matches the labelled reason type. Naming the type verbatim counts, and so does a general explanation that amounts to the same thing (see the description and examples given for the labelled type). A different reason, or a refusal with no reason, fails.
3. **No False Claim:** it does **NOT** claim to have performed, scheduled, or changed anything toward the request. Read-only exploration in the trace is fine.
4. **Consistency:** any factual claim it makes agrees with the tool observations in the trace.

Otherwise, output `B`.

DO NOT PENALIZE

- Which tools were used or how many steps were taken.
- How the answer is worded or how long it is.
- Whether the room or device names are mentioned.
- Extra suggestions or clarifying questions—as long as conditions 1–4 hold.

User Prompt

- **Person's request:** `{query}`
- **Labelled reason type:** `{reason_type}`
  - **Meaning:** `{reason_meaning}`
  - **Acceptable explanations sound like:** `{reason_examples}`
- **Ground-truth details (from the episode's evaluation spec):** `{goals}`
- **Agent's ReAct Steps (tool calls and observations):** `{react_steps}`
- **Agent's Final Answer:** `{final_answer}`

EVALUATION STEPS

1. Does the Final Answer state that the request cannot be / was not carried out?
2. Does the reason it gives match "`{reason_type}`" (verbatim, or a general explanation amounting to the meaning above)?
3. Does it avoid claiming to have performed or scheduled the requested change?
4. Are its factual claims consistent with the tool observations?

Pass only if all four hold.

Your Decision: `A or B`

Reason types. The judge is parameterised by the episode's labelled infeasibility type; `{reason_meaning}` and `{reason_examples}` are filled from the table below.

- `non_existent`: no device matching what the person referred to exists in the room: the named / described device is not there, no device is making the described sound, or there is no device in the described direction. *Examples:* "there is no such device in the room", "nothing here is making that sound", "I could not find any device in that direction".

- `non_actuator`: the person wants a **room** state changed (brighter / darker / warmer / cooler / more or less humid / cleaner air) but no device in that room can move that state. *Examples:* "there is no light in that room to make it brighter", "nothing in the room can cool it down".
- `physical_limit`: the target device exists and was correctly identified, but it is already at the limit of what was asked (e.g. fan speed already 100 percent, brightness already at maximum), so the requested change cannot go any further. *Examples:* "it is already at maximum", "the fan is already at 100 percent so it cannot go faster".
- `temporal_conflict`: the timing of the request is contradictory or impossible: the person stated two times that do not agree (e.g. "in 25 minutes, at 11:43" when those are different moments), or asked for an ordering of events that cannot hold. *Examples:* "the two times you gave do not match", "the schedule contradicts itself".

F Human Annotation

F.1 Human Dataset Verifier

We manually verified the dataset to ensure the quality and correctness of the collected examples. Human annotators inspected each example using the annotation interface shown in Fig. F.1, and checked whether the provided inputs and annotations were consistent with the intended task.

F.2 Human Performance

Evaluating human performance on agentic tasks is time-consuming, as solving each task requires multiple steps of environment exploration, reasoning, and interaction. Following prior agent benchmarks that estimate human performance on a representative subset of tasks Gou et al. (2025); Zhou et al. (2024); Lu et al. (2024); Chen et al. (2026), we evaluated human performance on a balanced sample of approximately 25% of the benchmark. We sampled 400 episodes in total and assigned 200 episodes to each of two human participants. Participants interacted with the environment through the human solver interface shown in Fig. F.2 and attempted to complete each task following the same task specification used for agent evaluation.

G Limitation

Our benchmark assumes a single-turn interaction setting and thus does not capture multi-turn interactions in which users may provide additional context, clarification, or corrections. In addition, in the synthetic portion of the benchmark, visual observations are provided as static images rather than videos, which limits the evaluation of temporally evolving visual information. On the method side, PROME is primarily designed to improve grounding and does not explicitly address other challenging aspects of embodied or smart-home agents, such as temporal scheduling and implicit intent reasoning. Finally, although we analyze the number of steps taken by the agent, we have not yet evaluated the latency introduced by perception-tool inference or the feasibility of real-time deployment. Addressing these limitations is an important direction for future work.

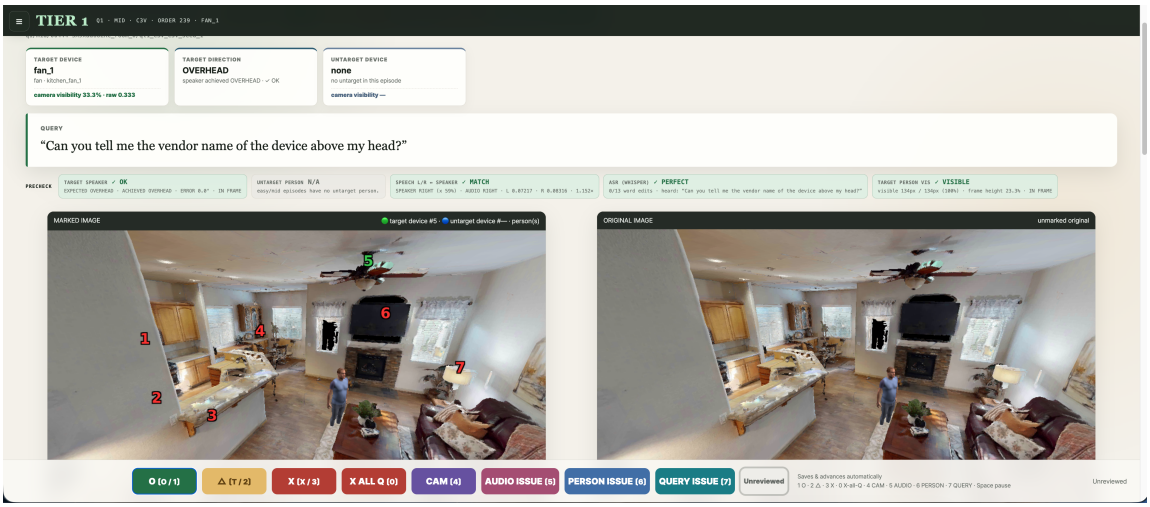


Figure F.1: Human verification interface used to manually inspect and validate dataset examples.

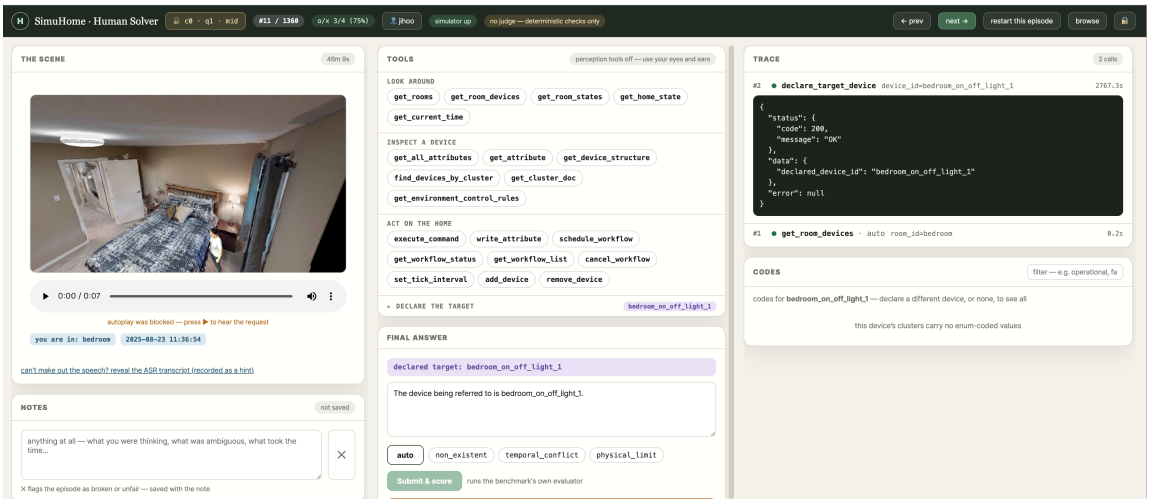


Figure F.2: Human solver interface used to evaluate human performance on our benchmark. Participants inspect the task observations and interact with the environment to complete the given user request.